Universität Bremen

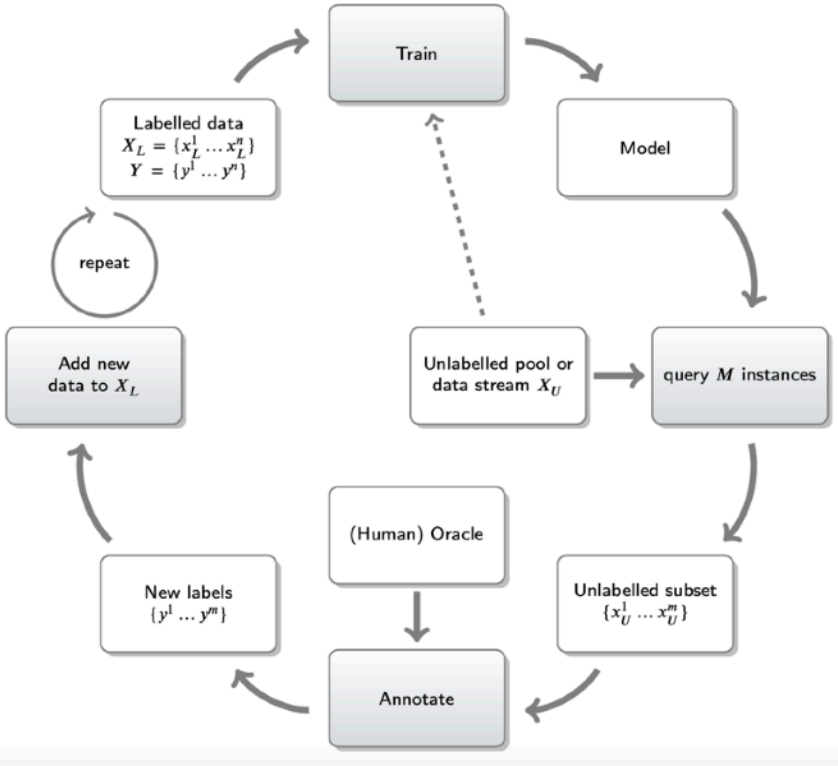


Figure from “A Wholistic View of Deep Neural Networks”, Neural Networks 160, Mundt et al 2023.

Part 3 - The Future

From Active Data Queries to Unknowns and Open Worlds

Lifelong Machine Learning
Summer 2025

Prof. Dr. Martin Mundt

Where do new data & patterns come from?

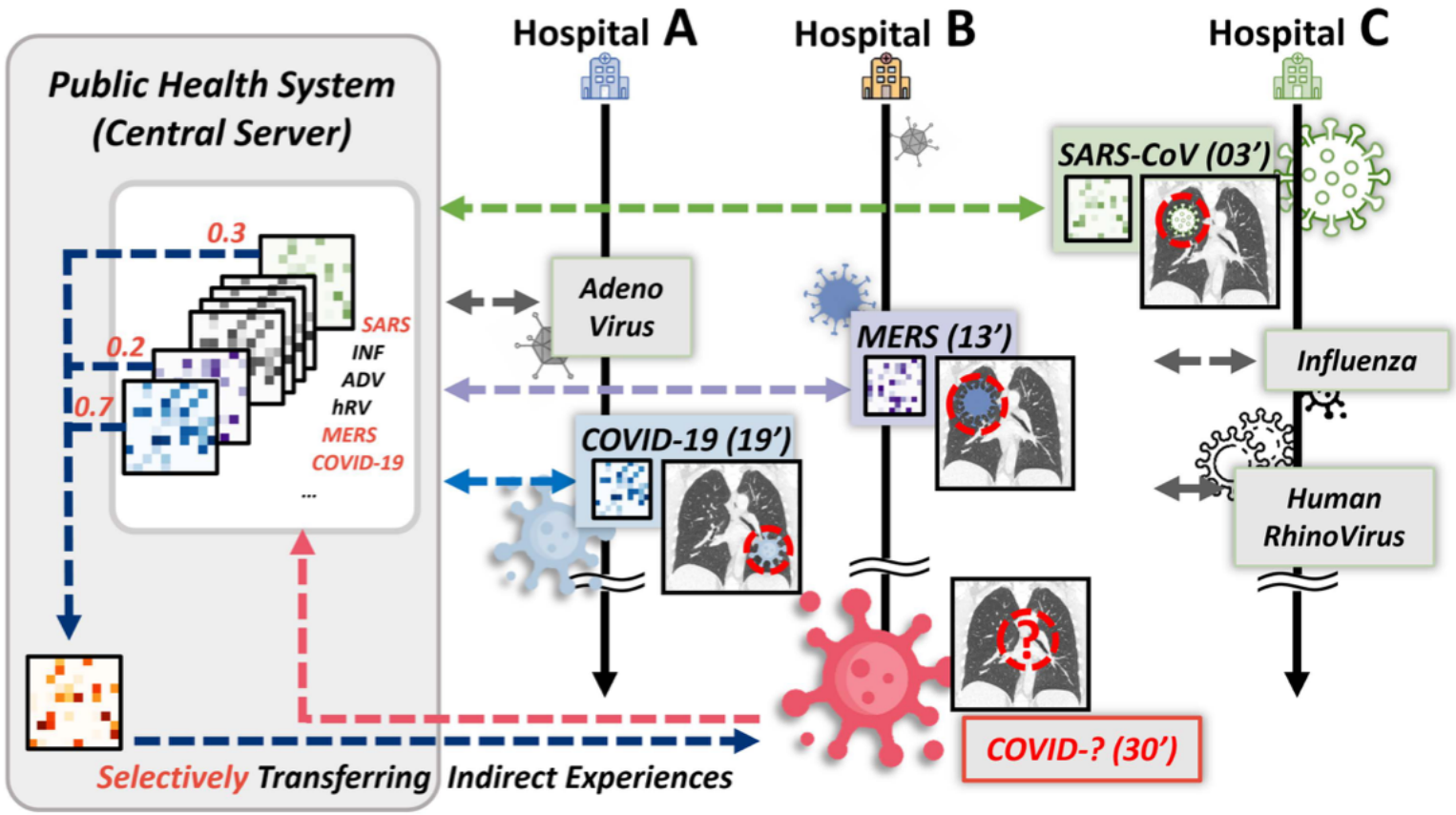


Image from Yoon et al, "Federated Continual Learning with Weighted Inter-client Transfer", ICML 2021

Universität Bremen

OWL ML

What data do we continue using for training?

As the 1st question “towards the future”, let’s consider the following:

We have started a training process on some initial data and we would now like to decide **what data to include next** into training

Why is this question valuable and what makes it interesting?

What data do we continue using for training?

As the 1st question “towards the future”, let’s consider the following:

We have started a training process on some initial data and we would now like to decide **what data to include next** into training

Why is this question valuable and what makes it interesting?

- Not all data is equally informative (recall rehearsal & coresets!)
- Data “is cheap” in comparison to labels/annotations! (Supervision)

Question time

You've trained on some data items and now wish to add more data to your training.
What important set-up assumptions can you make & how will they influence your prospective strategy?

Querying new data points and/or labels

Typically called **active learning** (sometimes “query” learning and related to “optimal experimental design” in statistics literature)

Requires us to think of an “**acquisition function**” to select data

Querying new data points and/or labels

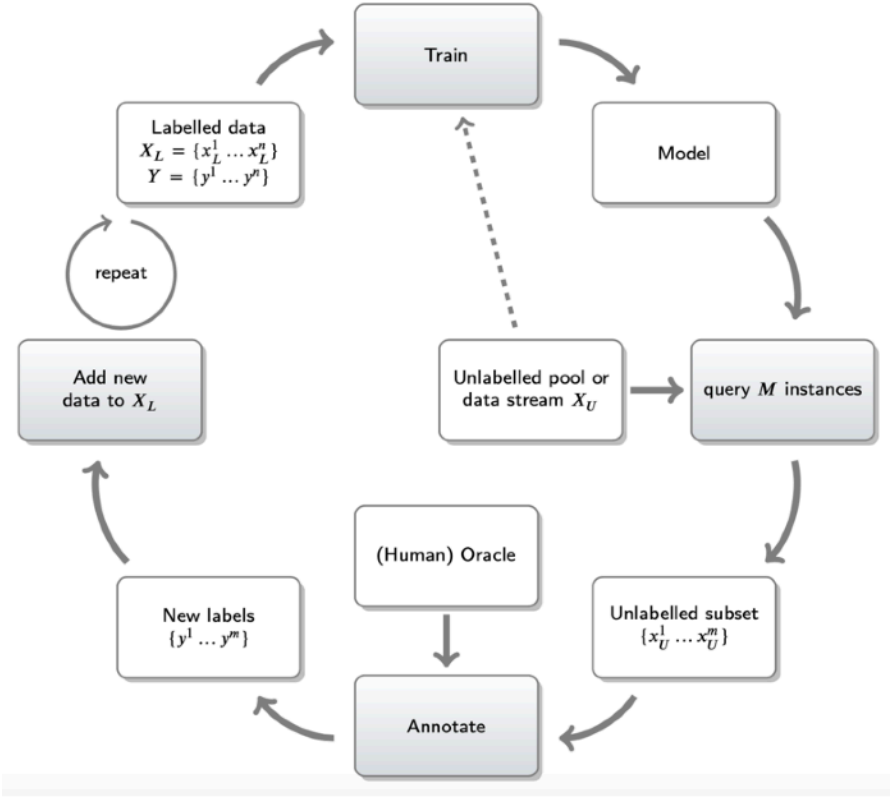
Typically called **active learning** (sometimes “query” learning and related to “optimal experimental design” in statistics literature)

Requires us to think of an “**acquisition function**” to select data

Requires us to think about whether we want to select **one datapoint** or a **batch of data** (or even an entire new task)

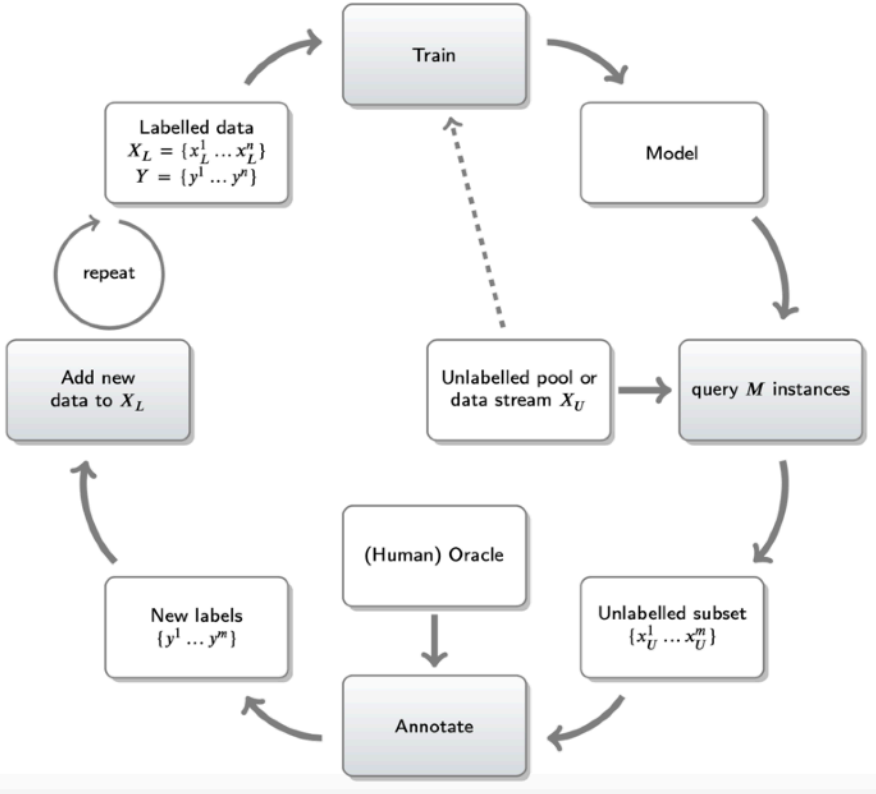
Involves some kind of (human) **oracle or teacher** to query for additional information, such as labels required for supervision

Active learning: pools & data accumulation



Left figure from “A Wholistic View of Deep Neural Networks: Forgotten Lessons and the Bridge to Active and Open World Learning”, Neural Networks 160, Mundt et al 2023.

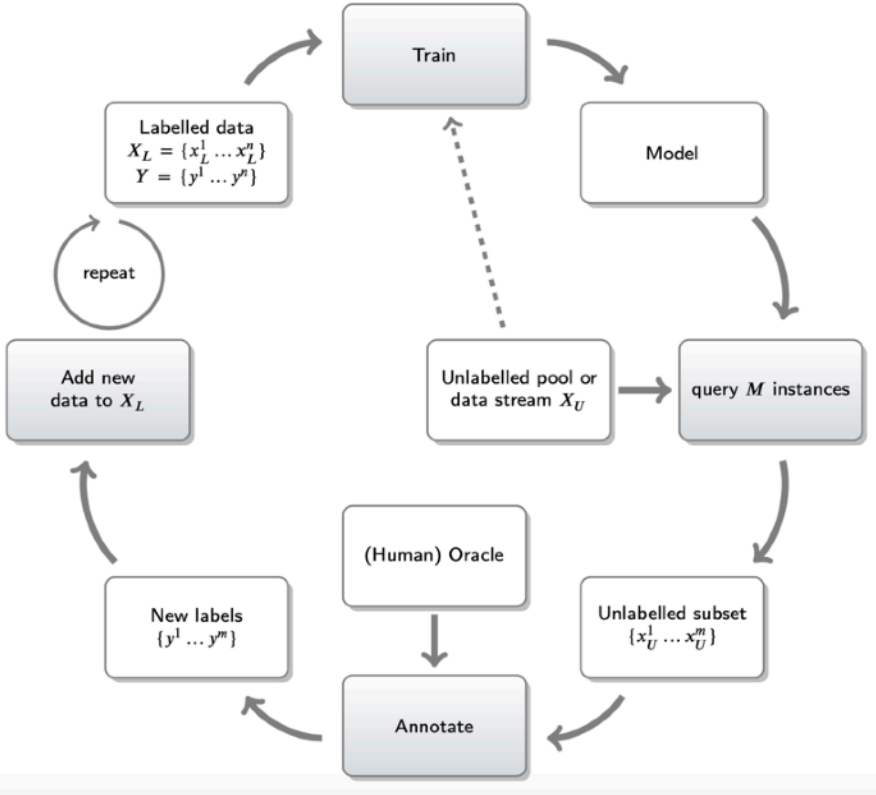
Active learning: pools & data accumulation



A BIG assumption:
Can we select data from a **pool**, or do we need to filter a **stream** of data?

Left figure from “A Wholistic View of Deep Neural Networks: Forgotten Lessons and the Bridge to Active and Open World Learning”, Neural Networks 160, Mundt et al 2023.

Active learning: pools & data accumulation



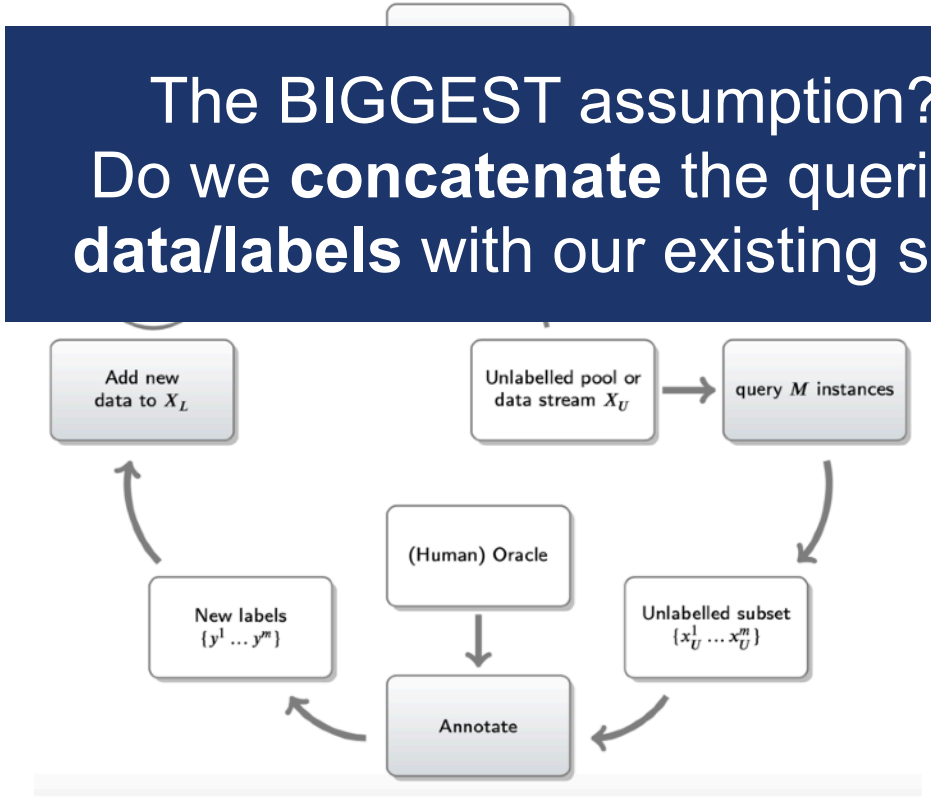
A BIG assumption:
Can we select data from a **pool**, or do we need to filter a **stream** of data?

An even BIGGER assumption:
Do we get **perfect labels** from the data from an **oracle** or is it **fallible**?

Left figure from “A Wholistic View of Deep Neural Networks: Forgotten Lessons and the Bridge to Active and Open World Learning”, Neural Networks 160, Mundt et al 2023.

Active learning: pools & data accumulation

The BIGGEST assumption?
Do we **concatenate** the queried **data/labels** with our existing set?



Given: Labeled set $\mathcal{L}$, unlabeled pool $\mathcal{U}$, query strategy $\phi(\cdot)$, query batch size B

repeat
  // learn a model using the current $\mathcal{L}$
  $\theta = \text{train}(\mathcal{L})$;
  for $b = 1$ **to** B **do**
    // query the most informative instance
    $\mathbf{x}_b^* = \arg\max_{\mathbf{x} \in \mathcal{U}} \phi(\mathbf{x})$;
    // move the labeled query from $\mathcal{U}$ *to* $\mathcal{L}$
    $\mathcal{L} = \mathcal{L} \cup \langle \mathbf{x}_b^*, \text{label}(\mathbf{x}_b^*) \rangle$;
    $\mathcal{U} = \mathcal{U} - \mathbf{x}_b^*$;
  end
until *some stopping criterion* ;

Algorithm 1: Pool-based active learning.

Left figure from “A Wholistic View of Deep Neural Networks: Forgotten Lessons and the Bridge to Active and Open World Learning”, Neural Networks 160, Mundt et al 2023. Right figure from Settles, "An Analysis of Active Learning Strategies for Sequence Labeling Tasks”, EMNLP 2008

Active learning

Traditionally, active learning can be pool or stream based.
However, “standard” active learning always **grows the dataset**.

In the course, we thus need to do a momentary mental step back.

Active learning

Traditionally, active learning can be pool or stream based.
However, “standard” active learning always **grows the dataset**.

In the course, we thus need to do a momentary mental step back.

Even more importantly, traditional active learning **concentrates exclusively on the value of our acquisition function**.

That is, we will query a data point (or a batch of data points), add to our dataset, and then **re-train the model from scratch** to compare different acquisition strategies and isolate other effects

Question time

We'll see that by itself, this is already challenging!
What techniques can you think of to acquire data?

Universität Bremen

Acquisition function perspectives

Uncertainty & heuristics

The intuitive approach: use the predictions, or maybe even better, uncertainty in the predictions for the queries

Acquisition function perspectives

Uncertainty & heuristics

The intuitive approach: use the predictions, or maybe even better, uncertainty in the predictions for the queries

Representation learning (& coresets)

The distribution based approach: maximizing coverage instead of reducing the possible set of hypotheses (version space) explicitly

Acquisition function perspectives

Version space reduction

The formal approach: reduce the set/space of possible hypotheses: $h : \mathcal{X} \to \mathcal{Y}$ by removing ones that are inconsistent with the data

Uncertainty & heuristics

The intuitive approach: use the predictions, or maybe even better, uncertainty in the predictions for the queries

Representation learning (& coresets)

The distribution based approach: maximizing coverage instead of reducing the possible set of hypotheses (version space) explicitly

Universität Bremen

OWL ML

Question time

Should we make us of discriminative or generative models for active learning? What are pros & cons?

Active learning: discriminative or generative?

Discriminative model based active learning
could allow for natural ways to assess “novelty” of a new example

Generative model based active learning
could allow for natural ways to assess similarity to data distribution

Active learning: discriminative or generative?

Discriminative model based active learning

could allow for natural ways to assess “novelty” of a new example

-> *Caution*: overconfidence phenomena (next lecture)

Generative model based active learning

could allow for natural ways to assess similarity to data distribution

-> *Caution*: our parameters only reflect the distribution seen so far!

We will see that model type utility will depend heavily on our choices in terms of set-up assumptions, e.g. are we using a pool of data

Defining the version space

Assume that there exist hypotheses consistent with labeled data $h : \mathcal{X} \rightarrow \mathcal{Y}$, then the **version space** is defined as:

$$VS(D) = \{h \in H \,|\, \texttt{cons}(h, D)\}$$

Defining the version space

Assume that there exist hypotheses consistent with labeled data $h : \mathcal{X} \rightarrow \mathcal{Y}$, then the **version space** is defined as:

$$VS(D) = \{h \in H \,|\, \texttt{cons}(h, D)\}$$

- **Specific hypotheses**: cover positive examples & as little remaining feature space as possible
- **General hypotheses**: cover positive examples & as much of the remaining feature space as possible

Defining the version space

Assume that there exist hypotheses consistent with labeled data $h : \mathcal{X} \rightarrow \mathcal{Y}$, then the **version space** is defined as:

$$VS(D) = \{h \in H \mid \texttt{cons}(h, D)\}$$

- **Specific hypotheses**: cover positive examples & as little remaining feature space as possible
- **General hypotheses**: cover positive examples & as much of the remaining feature space as possible

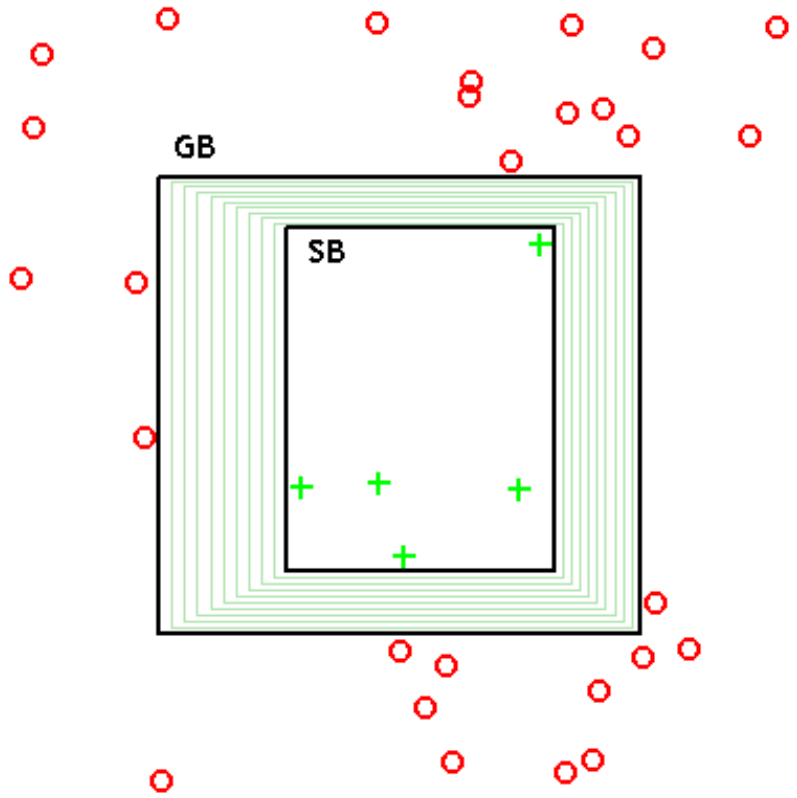


Figure from https://en.wikipedia.org/wiki/File:Version_space.png in the public domain

Defining the version space

Assume that there exist hypotheses consistent with labeled data $h : \mathcal{X} \rightarrow \mathcal{Y}$, then the **version space** is defined as:

$$VS(D) = \{h \in H \mid \texttt{cons}(h, D)\}$$

- **Specific hypotheses**: cover positive examples & as little remaining feature space as possible
- **General hypotheses**: cover positive examples & as much of the remaining feature space as possible

Version space: space "between" general & specific hypothesis. — here, green rectangles

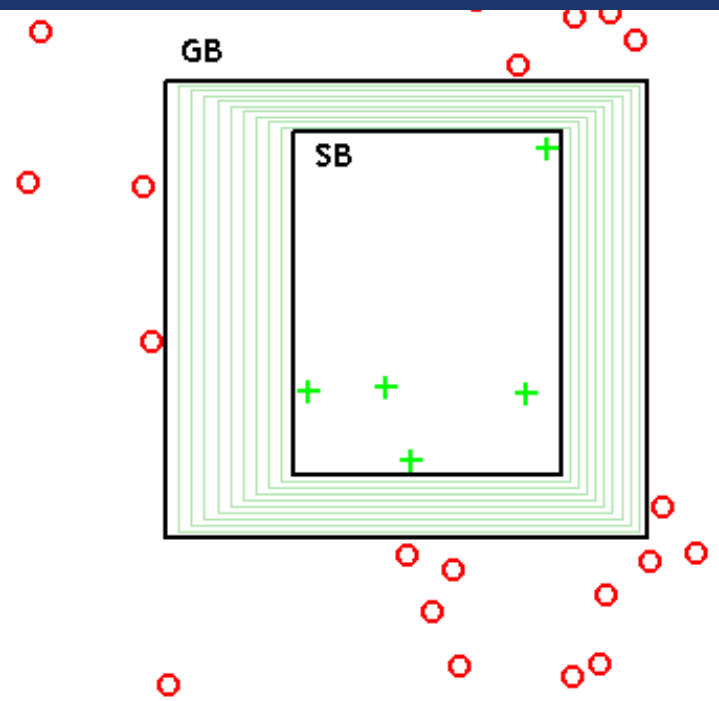


Figure from https://en.wikipedia.org/wiki/File:Version_space.png in the public domain

“Generalization as Search” (Mitchell 1982)

If we want to think about how the version space is interesting for active learning then:

We could query such that the **version space**:

$VS(D) = \{h \in H \mid \texttt{cons}(h, D)\}$

reduces the version space quickly

“Generalization as Search” (Mitchell 1982)

If we want to think about how the version space is interesting for active learning then:

We could query such that the **version space**:
$VS(D) = \{h \in H \,|\, \texttt{cons}(h, D)\}$
reduces the version space quickly

We get a smaller prospective set of consistent hypotheses

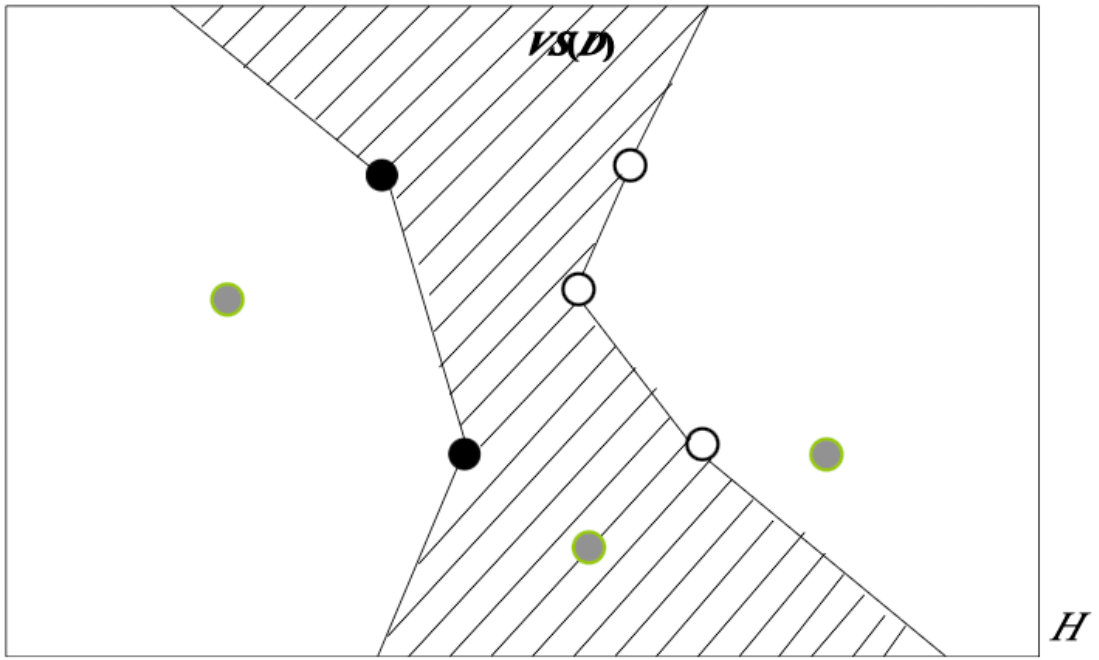


Figure from presentation of “Ensembles of Classifiers” by Evgueni Smirnov, slides: https://slideplayer.com/slide/10075963/

Question time

Version spaces seem to be hard to obtain in practice - do you recall any models where we could perhaps define a version space?

A short excursion: support vector machines

SVMs are very interesting for active learning with version spaces

What is a SVM?

- Not too different from logistic regression, NNs etc.
- Choose hyperplane that divides data points into the two classes

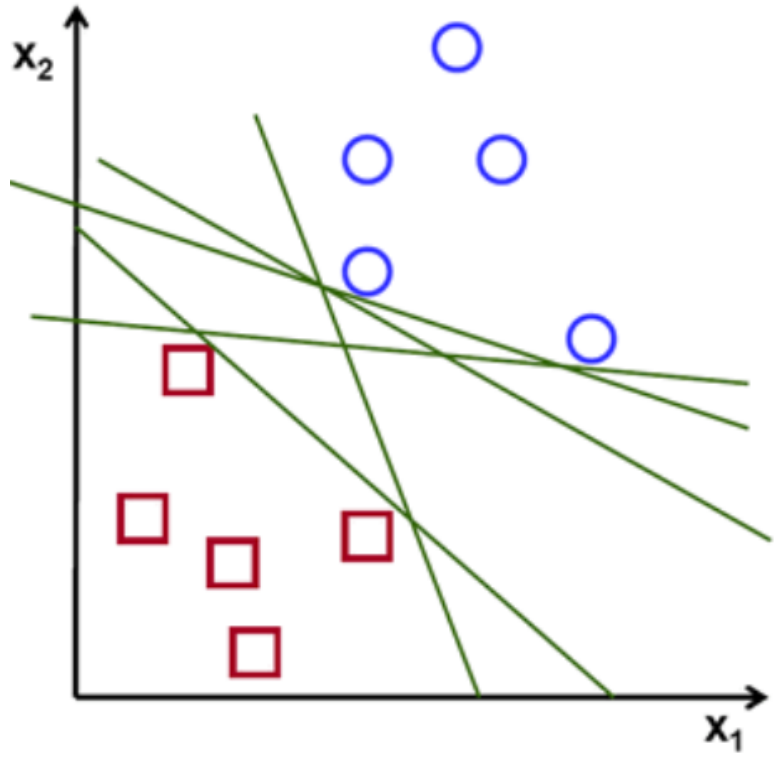


Figure from https://towardsdatascience.com/support-vector-machine-vs-logistic-regression-94cc2975433f

A short excursion: support vector machines

SVMs are very interesting for active learning with version spaces

What is a SVM?

- Not too different from logistic regression, NNs etc.
- Choose hyperplane that divides data points into the two classes

Do you recall how SVMs work?

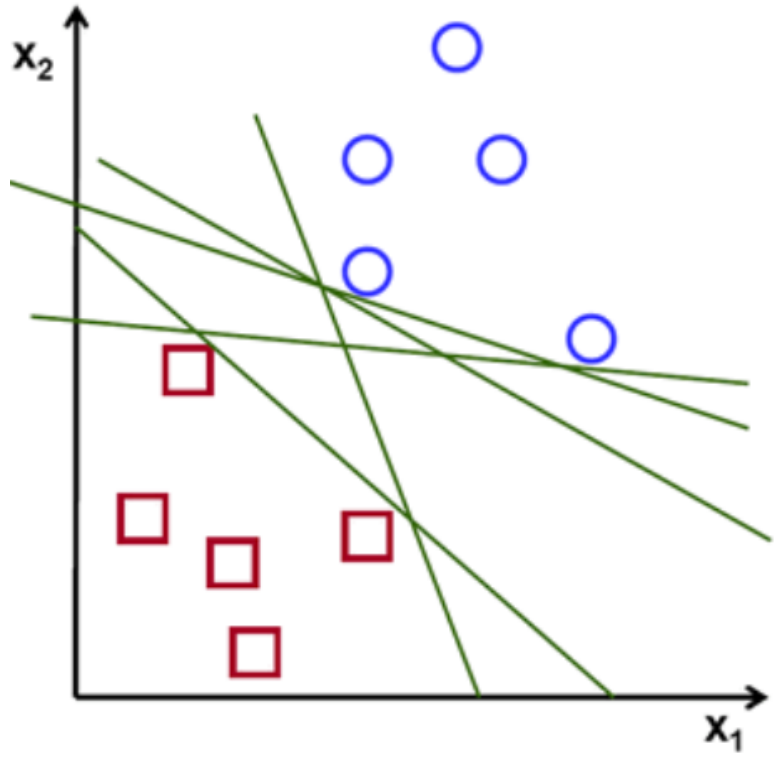


Figure from https://towardsdatascience.com/support-vector-machine-vs-logistic-regression-94cc2975433f

A short excursion: support vector machines

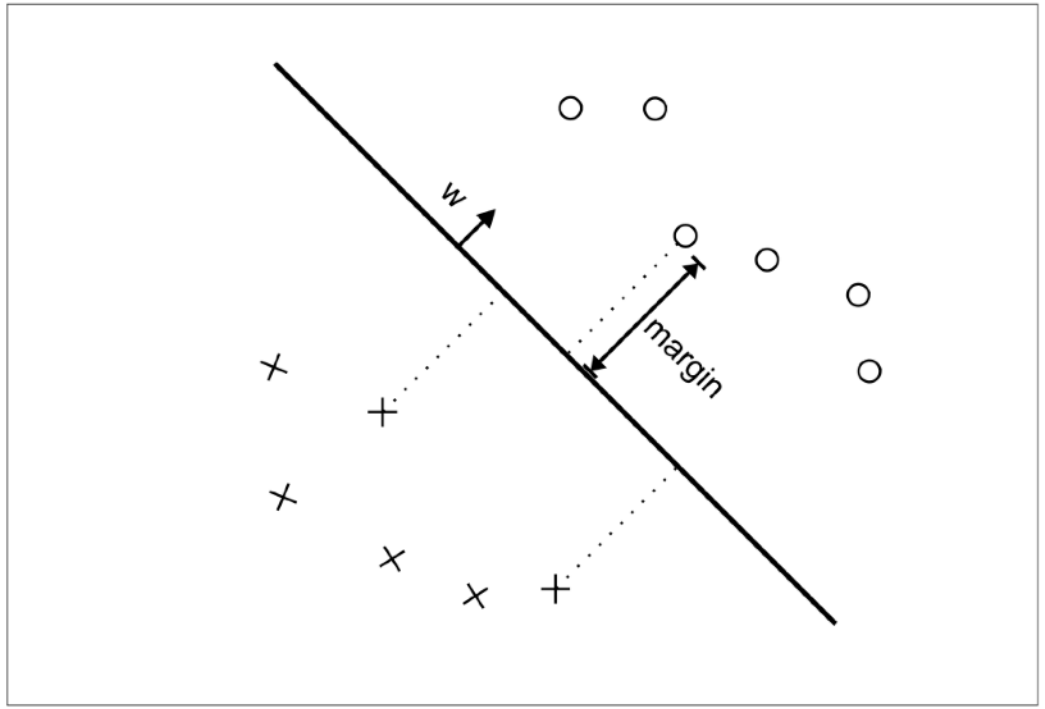
Any hyperplane can be written as a set of points x satisfying:

$$w^T x - b = 0$$

where **w** is the **normal vector**

Margin:

$$w^T x - b = 1 \;\&\; w^T x - b = -1$$



Tong & Koller, “Support Vector Machine Active Learning with Applications to Text Classification”, JMLR 2001

A short excursion: support vector machines

Hyperplane chosen to **maximize margin to closest instances**: the support vectors

- We can write:
  $y_i(w^T x_i - b) = 0 \geq 1, \forall 1 \leq i \leq n$
  (additionally, no points fall on the boundary)

- Optimization problem — **minimize ||w||** subject to:
  $y_i(w^T x_i - b) = 0 \geq 1, \forall 1 \leq i \leq n$

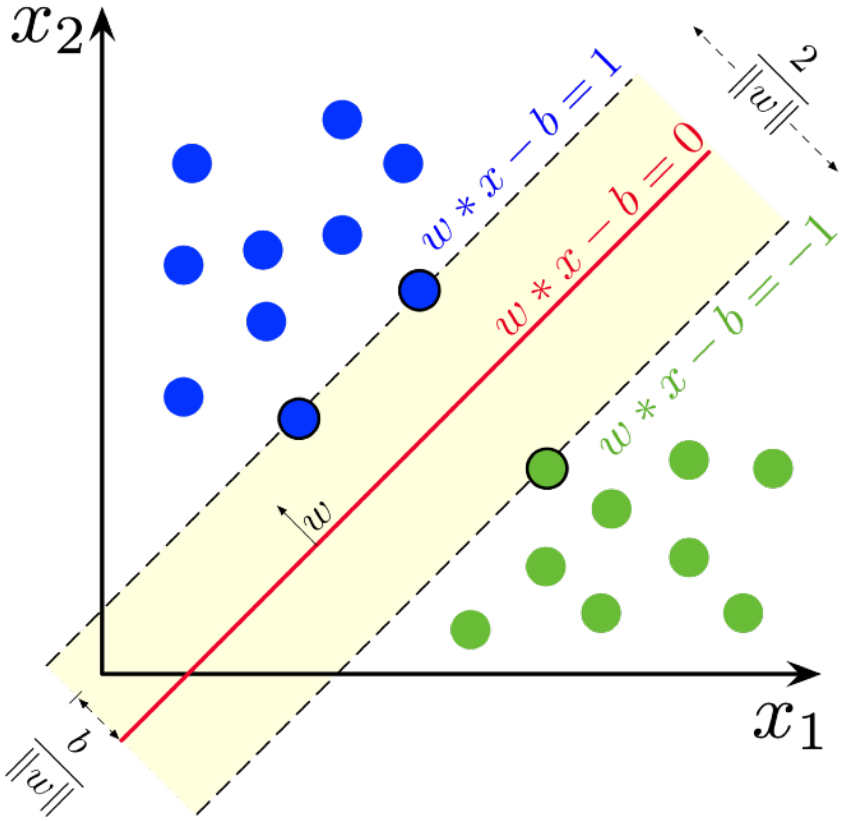


Figure from https://en.wikipedia.org/wiki/File:SVM_margin.png shared under CC 4.0 license

Universität Bremen

A short excursion: support vector machines

But our data may not always be linearly separable

Recall transfer learning: what can we do about this?

A short excursion: support vector machines

But our data may not always be linearly separable

Recall transfer learning: what can we do about this?

We can project data to a (higher dimensional) feature space

What is an intuitive transformation in the example?

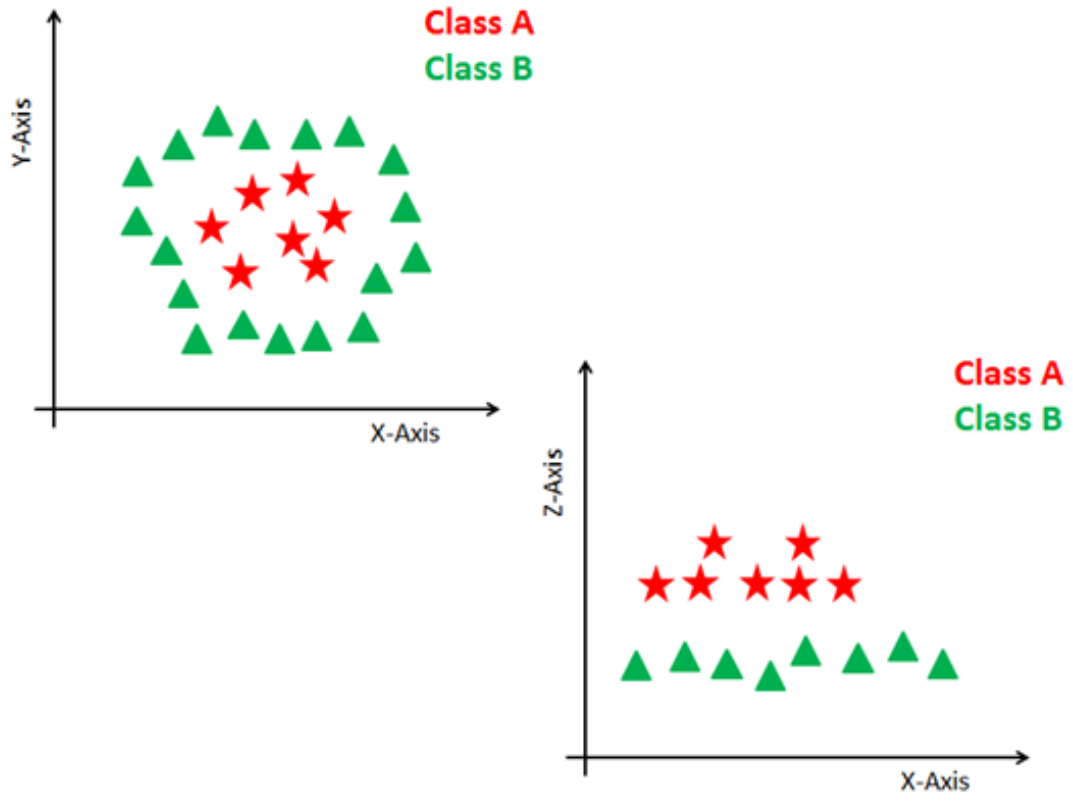


Figure from https://www.datacamp.com/tutorial/svm-classification-scikit-learn-python

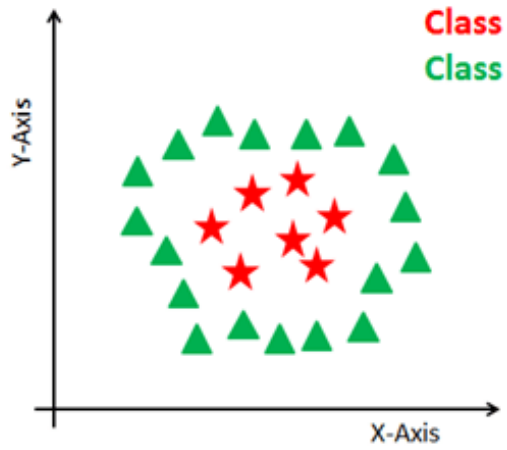
A short excursion: support vector machines

But our data may not always be linearly separable

Recall transfer learning: what can we do about this?

We can project data to a (higher dimensional) feature space

What is an intuitive transformation in the example?



polar coordinates

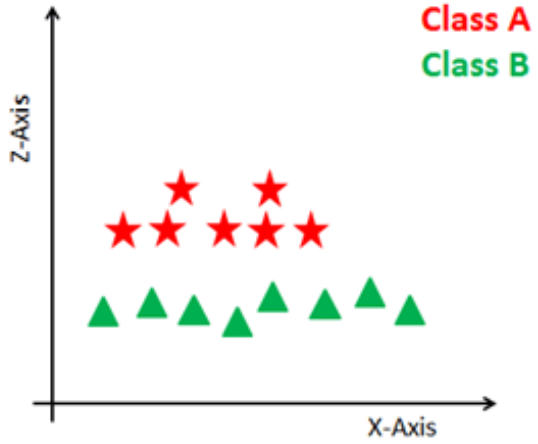
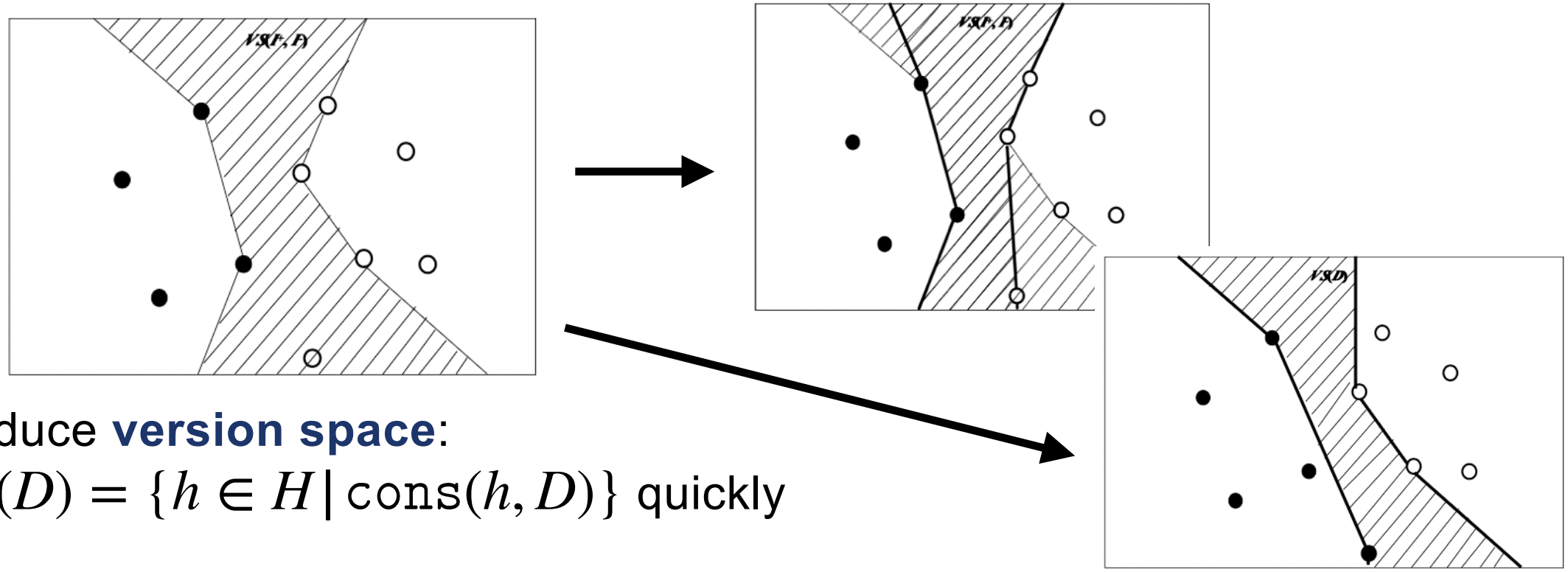


Figure from https://www.datacamp.com/tutorial/svm-classification-scikit-learn-python

Back to version spaces: specific for SVMs



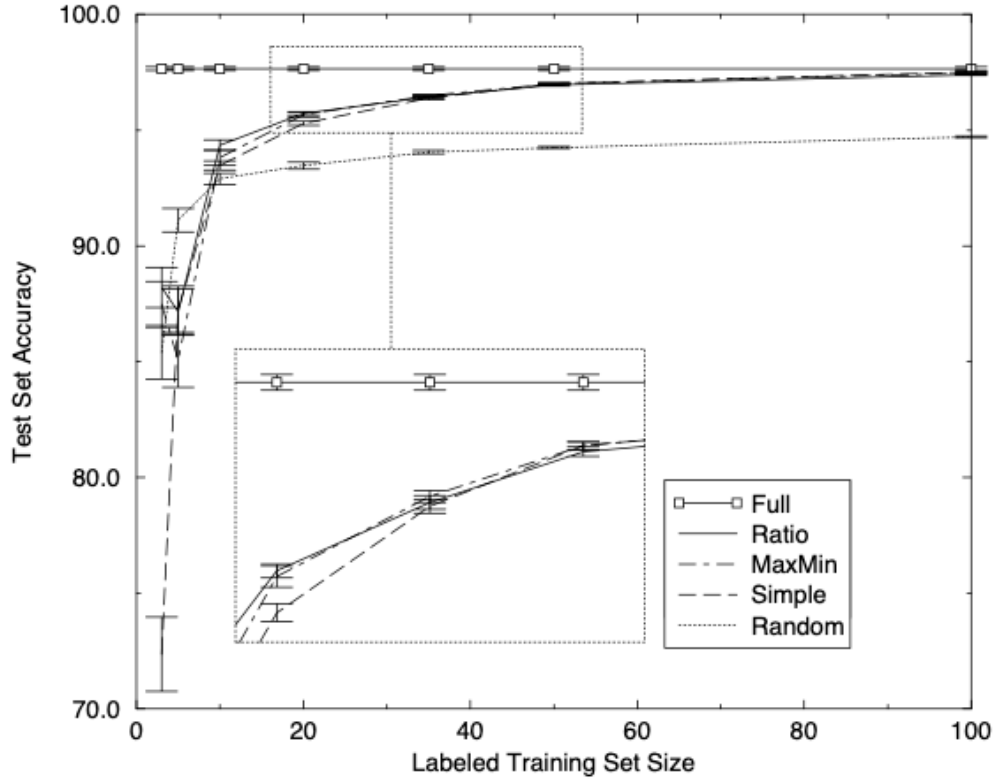
Reduce **version space**:
$VS(D) = \{h \in H \,|\, \texttt{cons}(h, D)\}$ quickly

Version space is **set of hyperplanes** (or could be redefined through vectors W)

Figure from presentation of “Ensembles of Classifiers” by Evgueni Smirnov, slides: https://slideplayer.com/slide/10075963/

Back to version spaces: specific for SVMs

An example on growing a standard benchmark dataset (with retraining!)



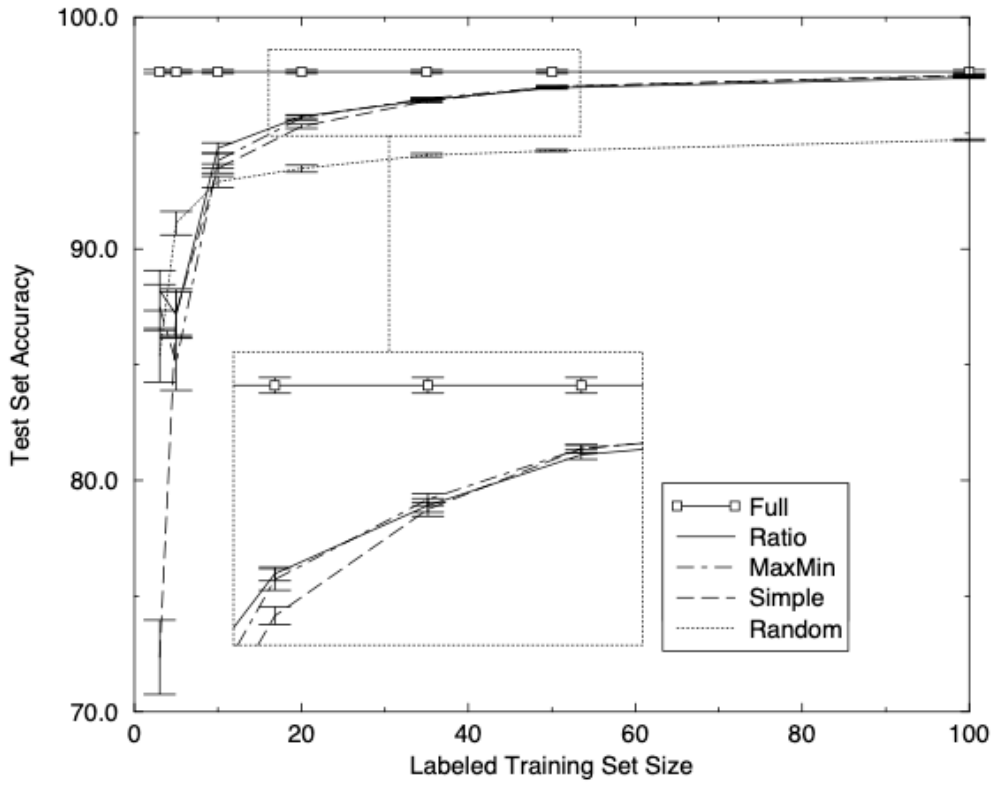
Tong & Koller, “Support Vector Machine Active Learning with Applications to Text Classification”, JMLR 2001

Back to version spaces: specific for SVMs

An example on growing a standard benchmark dataset (with retraining!)

Various approx.: is version space symmetric? Estimation of size?

What would be a solid strategy that you know from discussing search algorithms in CS theory?



Tong & Koller, “Support Vector Machine Active Learning with Applications to Text Classification”, JMLR 2001

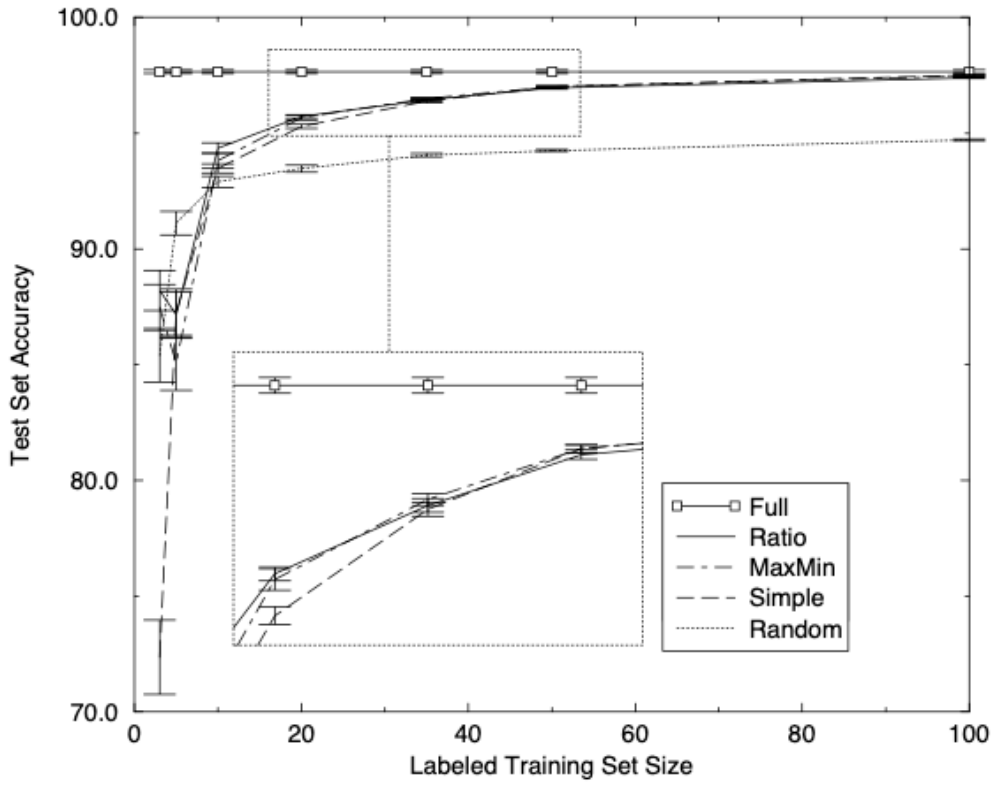
Back to version spaces: specific for SVMs

An example on growing a standard benchmark dataset (with retraining!)

Various approx.: is version space symmetric? Estimation of size?

What would be a solid strategy that you know from discussing search algorithms in CS theory?

Intuitively: choose successive queries that halve the version space



Tong & Koller, “Support Vector Machine Active Learning with Applications to Text Classification”, JMLR 2001

Universität Bremen

OWL ML

Question time

Reducing the set of hypotheses is interesting but often hard in practice - what other crucial quantity could we reduce that may be easier to handle?

An alternative to version spaces

Version space reduction (reducing the set of consistent hypothesis) disregards the **evaluation metric**!

An alternative to version spaces

Version space reduction (reducing the set of consistent hypothesis) disregards the **evaluation metric**!

We could also take a look at the **loss** & include points that would:

- most reduce the expected error
- most change the current model

An alternative to version spaces

Version space reduction (reducing the set of consistent hypothesis) disregards the **evaluation metric**!

We could also take a look at the **loss** & include points that would:

- most reduce the expected error
- most change the current model

"First-order Markov active learning aims to select a query $x^\star$, such that when the query is given label $y^\star$ & added to the training set, the learner trained on the resulting set $D+(x^\star,y^\star)$ has lower error than any other x"

Roy & McCallum, "Toward Optimal Active Learning through Monte Carlo Estimation of Error Reduction", ICML 2001) (A Cohn et al, "Active learning with statistical models", JAIR 4, 1996)

The simplest (?) approach to active learning

Version spaces & expected error reduction can be complicated & computationally heavy. Heuristics are thus popular (in deep learning)

1. Create an initial classifier
2. While teacher is willing to label examples
   (a) Apply the current classifier to each unlabeled example
   (b) Find the b examples for which the classifier is least certain of class membership
   (c) Have the teacher label the subsample of b examples
   (d) Train a new classifier on all labeled examples

Figure 1. An algorithm for uncertainty sampling with a single classifier.

Lewis & Gale, “A Sequential Algorithm for Training Text Classifiers”, ACM-SIGIR conference on research and development in information retrieval 1994

Information theory inspired quantities

Instead of pure output confidence, we could resort to information theory

Example: maximize expected information gain by querying examples with **largest entropy** (as a measure of disorder & related to information gain)

$$H(p) = -\sum_{i}^{c} p_i \log_2(p_i)$$

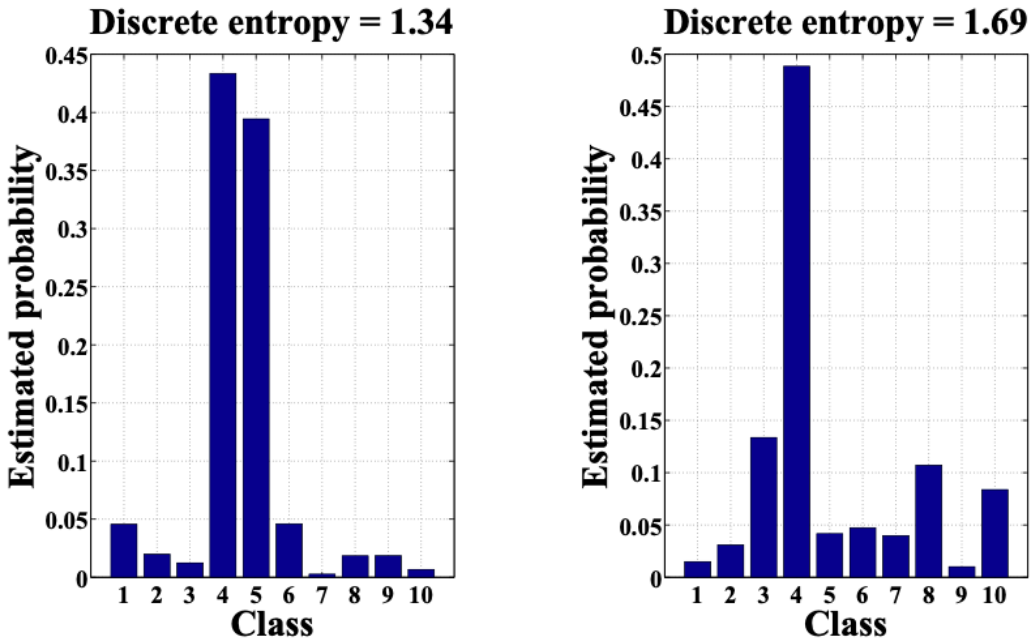
Example p(y|x):

- H[1.0, 0.0, 0.0, 0.0, 0.0] = 0
- H[0.2, 0.2, 0.2, 0.2, 0.2] = 1

See McKay, “Information-Based Objective Functions for Active Data Selection”, Neural Computation 4, 1992 based on prior works by Shannon 1948

Best versus second best

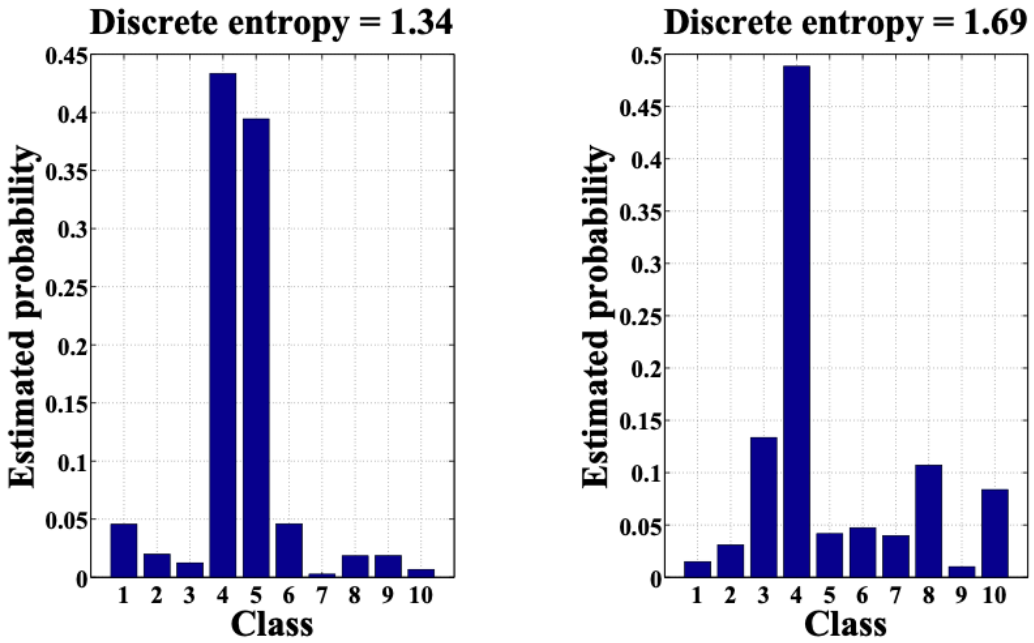
But entropy can also be a poor estimate when multiple classes are considered



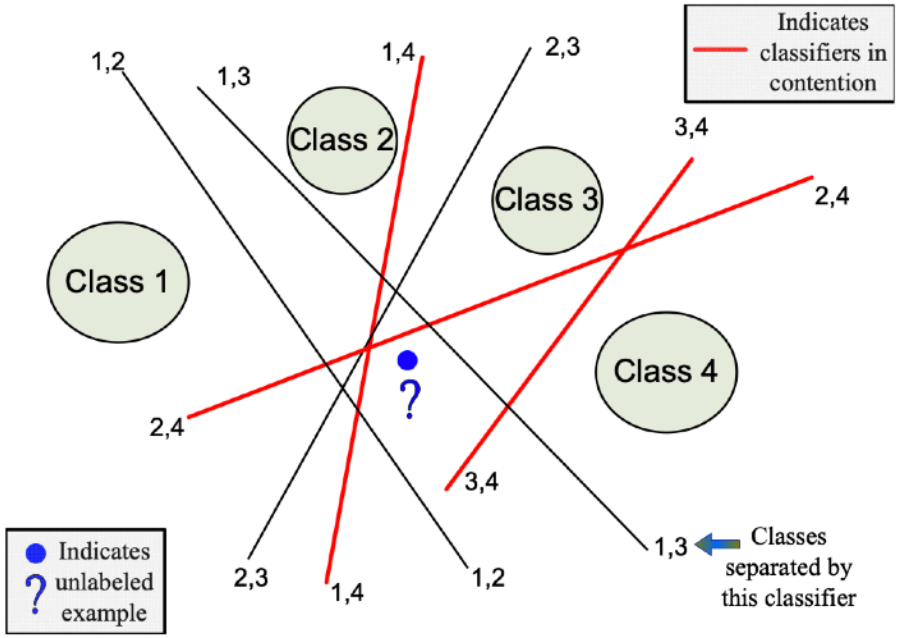
Joshi et al, “Multi-Class Active Learning for Image Classification”, CVPR 2009

Best versus second best

But entropy can also be a poor estimate when multiple classes are considered



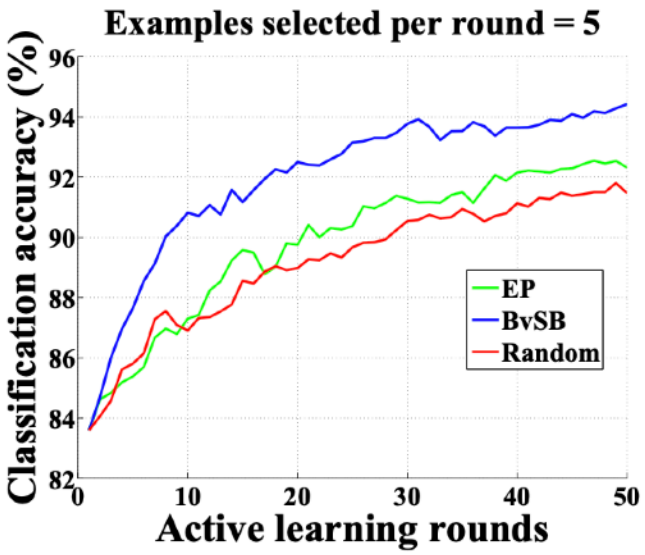
And (binary) classifiers can be in contention (e.g. in SVMs)

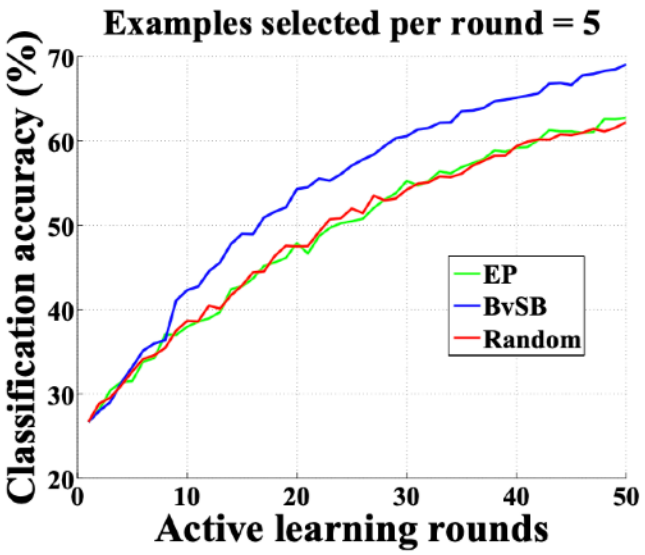


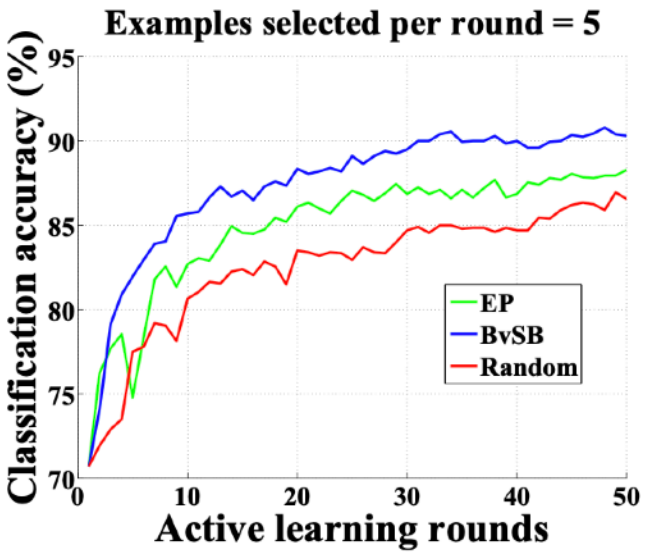
Joshi et al, “Multi-Class Active Learning for Image Classification”, CVPR 2009

Best versus second best

Left to right: Pendigits, Letter, USPS datasets







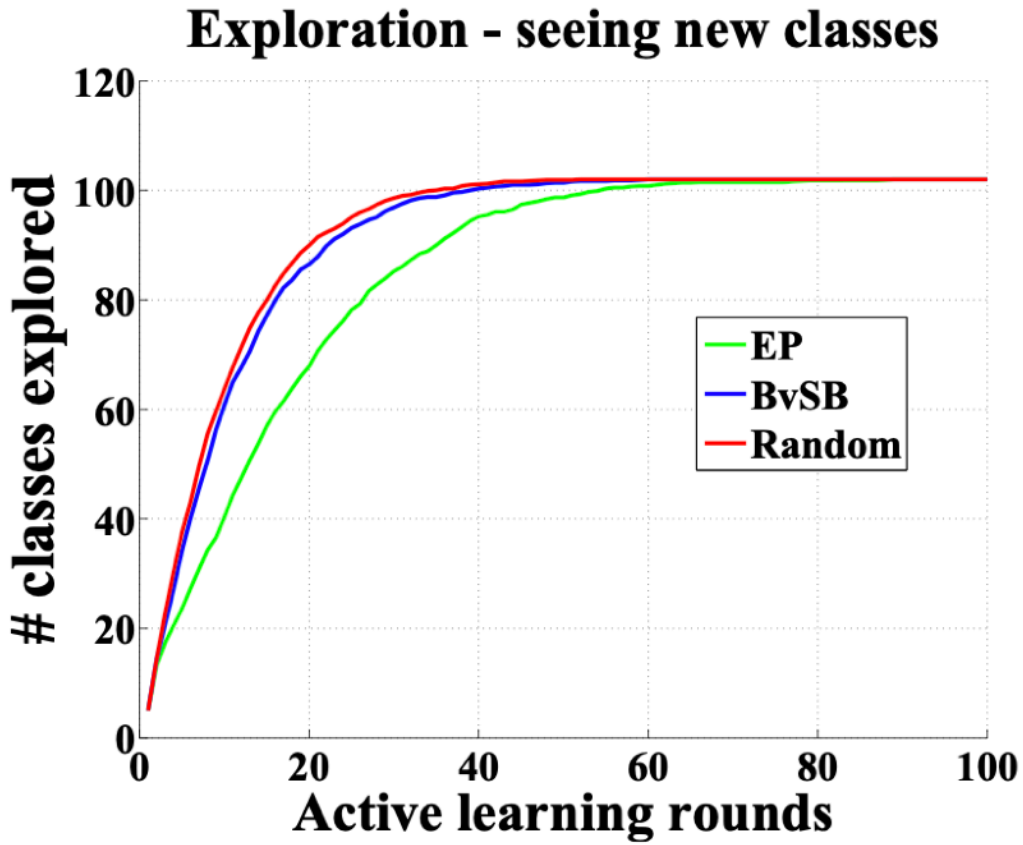
Joshi et al, “Multi-Class Active Learning for Image Classification”, CVPR 2009

Exploration versus exploitation

When the task isn't binary classification, we need to care about **exploration versus exploitation**

How much do we explore very novel classes & how much do we extend knowledge of what we have seen?

Our prediction-based measures typically <u>overemphasize "novelty"</u>



Joshi et al, "Multi-Class Active Learning for Image Classification", CVPR 2009

Can we correct entropy's novelty focus?

We could weigh entropy with some measure of data similarity in order to get an estimate of useful "**information density**":
(Settles, An Analysis of Active Learning Strategies for Sequence Labeling Tasks, EMNLP 2008)

$$ID(x) = -\sum_{\hat{y}} p(\hat{y}\,|\,x;\theta)\,\log p(\hat{y}\,|\,x;\theta) \cdot \frac{1}{U}\left[\sum_{u} \texttt{sim}(x, x^{(u)})\right]^{\beta}$$

Where beta is a weighting & the similarity over all unlabelled examples

U could be a distance: $\texttt{sim}_{cos}(x, x^{(u)}) = \dfrac{\vec{x} \cdot \vec{x}^{(u)}}{||\vec{x}|| \times ||\vec{x}^{(u)}||}$

Ensembles & prediction uncertainty

We could also maximize the information gain between two/multiple models: **ensembles**

Query by a committee of two
Repeat the following until n queries have been accepted

1. Draw an unlabeled input $x \in X$ at random from $\mathcal{D}$.
2. Select two hypotheses h_1, h_2 from the posterior distribution. In other words, pick two hypotheses that are consistent with the labeled examples seen so far.
3. If $h_1(x) \neq h_2(x)$ then query the teacher for the label of x, and add it to the training set.

Seung et al, “Query by Committee”, COLT 1992, and Freund, Seung et al, “Information, Prediction, and Query by Committee”, NeurIPS 1992

Ensembles & prediction uncertainty

We could also maximize the information gain between two/multiple models: **ensembles**

Do you see any relationship with version spaces here?

Query by a committee of two
Repeat the following until n queries have

1. Draw an unlabeled input $x \in X$ at random from $\mathcal{D}$.
2. Select two hypotheses h_1, h_2 from the posterior distribution. In other words, pick two hypotheses that are consistent with the labeled examples seen so far.
3. If $h_1(x) \neq h_2(x)$ then query the teacher for the label of x, and add it to the training set.

Seung et al, “Query by Committee”, COLT 1992, and Freund, Seung et al, “Information, Prediction, and Query by Committee”, NeurIPS 1992

Ensembles & prediction uncertainty

We could also maximize the information gain between two/multiple models: **ensembles**

Query by a committee of two
Repeat the following until n queries have

1. Draw an unlabeled input $x \in X$ at rand
2. Select two hypotheses h_1, h_2 from the pos pick two hypotheses that are consistent far.
3. If $h_1(x) \neq h_2(x)$ then query the teache the training set.

Do you see any relationship with version spaces here?

We can view the ensemble (if sufficiently large) as either an interpretation of a version space & reducing it, or as gauging a model's epistemic uncertainty

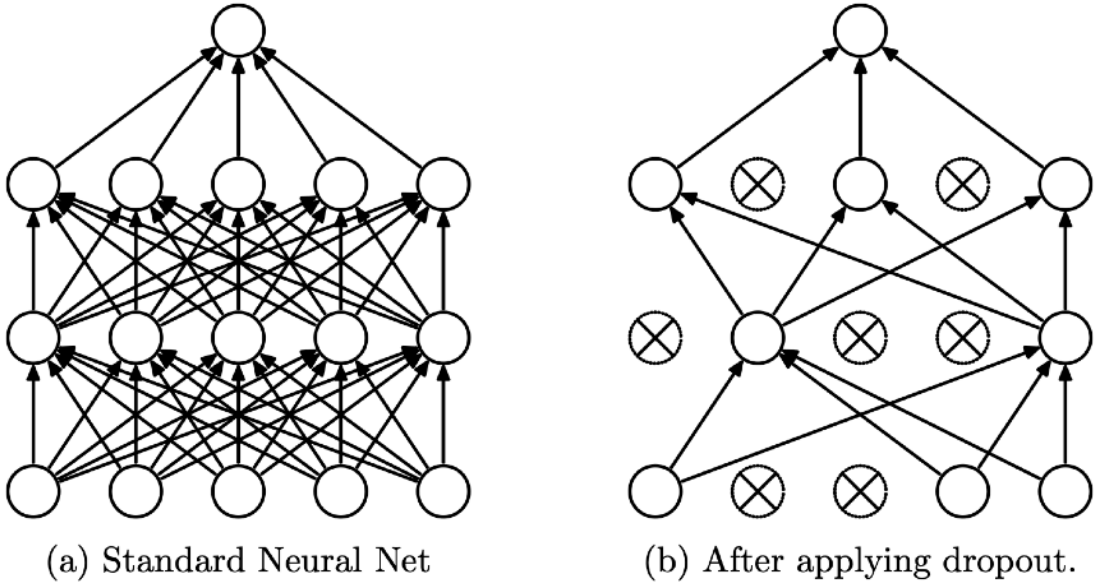
Seung et al, "Query by Committee", COLT 1992, and Freund, Seung et al, "Information, Prediction, and Query by Committee", NeurIPS 1992

Ensembles & prediction uncertainty

Ensembles are computationally VERY heavy & involve lots of training!

Monte Carlo Dropout (Gal et al, "Dropout as a Bayesian Approximation", ICML 2016)

- Make use of dropout: randomly turning off units in a model
- Bayesian interpretation: Bernoulli distribution on the parameters
- Do stochastic forward passes to assess variation in predictions (model uncertainty)



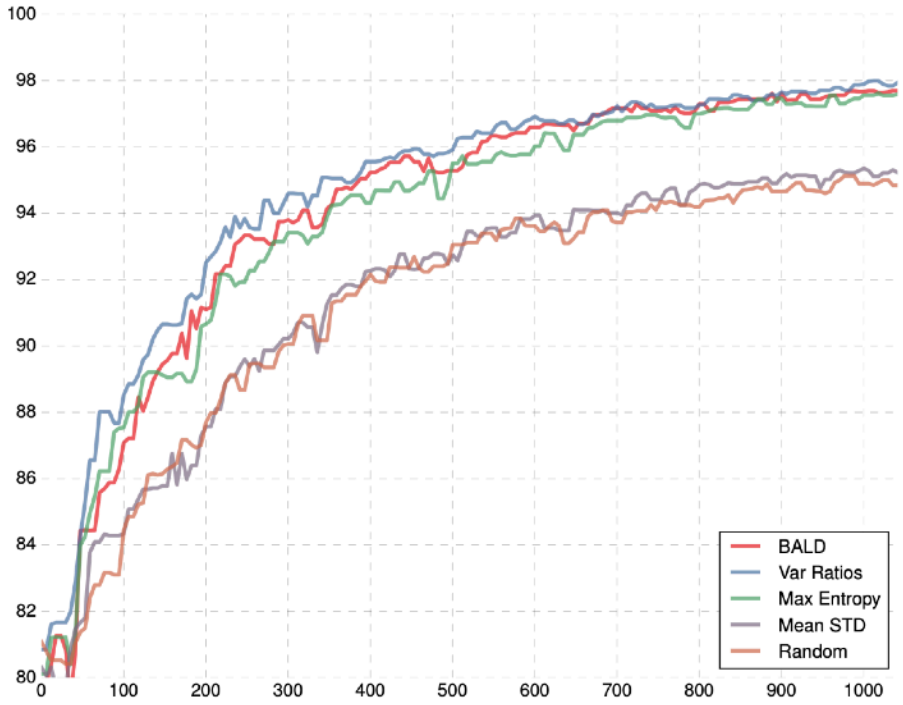
(a) Standard Neural Net (b) After applying dropout.

Srivastava et al, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", JMLR 15, 2014

Ensembles & prediction uncertainty

MCD could be useful as an approximation to using multiple model based ensembles

The acquisition function could still be entropy, standard deviation in output confidence etc.



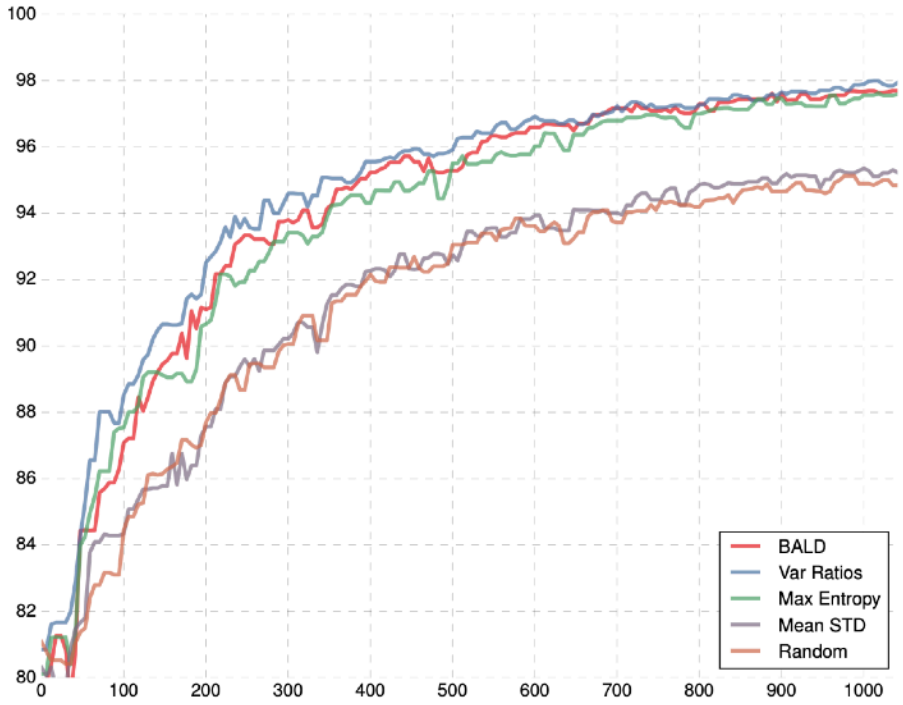
Gal et al, “Deep Bayesian Active Learning with Image Data”, ICML 2017

Ensembles & prediction uncertainty

MCD could be useful as an approximation to using multiple model based ensembles

The acquisition function could still be entropy, standard deviation in output confidence etc.

Active queries based on confidence, entropy, and Monte Carlo dropout sampling will be the final tutorial



Gal et al, “Deep Bayesian Active Learning with Image Data”, ICML 2017

Universität Bremen

OWL ML

Question time

Are you at all surprised to see the experimental results? Why aren't these strategies a lot better in contrast to a random uniform sampling baseline?

Limits of predictions & sampled uncertainty

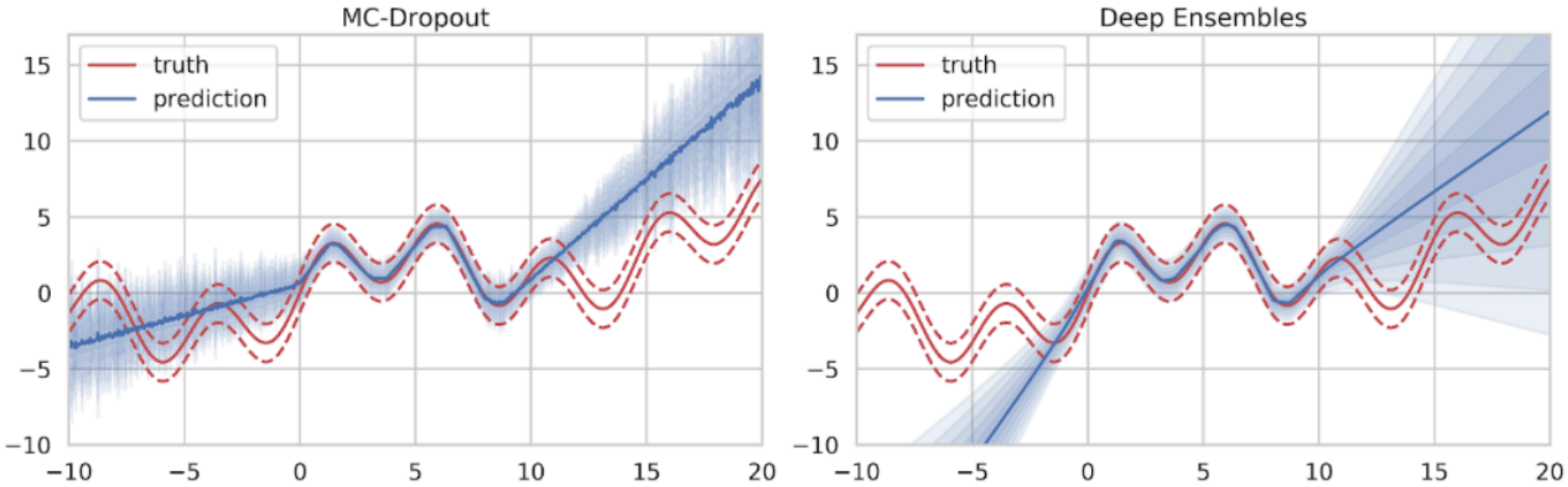


Figure from https://www.inovex.de/de/blog/uncertainty-quantification-deep-learning/

Universität Bremen

Limits of predictions & sampled uncertainty

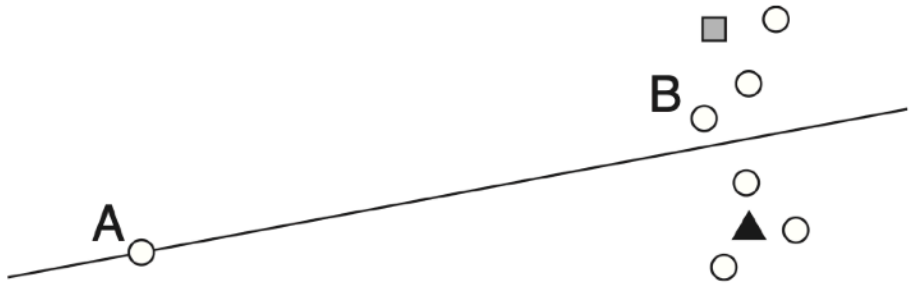


Figure 2: An illustration of when uncertainty sampling can be a poor strategy for classification. Shaded polygons represent labeled instances ($\mathcal{L}$), and circles represent unlabeled instances ($\mathcal{U}$). Since A is on the decision boundary, it will be queried as the most uncertain. However, querying B is likely to result in more information about the data as a whole.

Settles & Craven, "An Analysis of Active Learning Strategies for Sequence Labeling Tasks", EMNLP 2008

Limits of predictions & sampled uncertainty

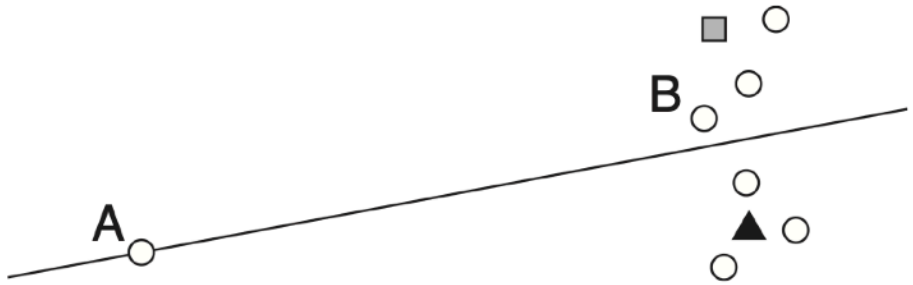


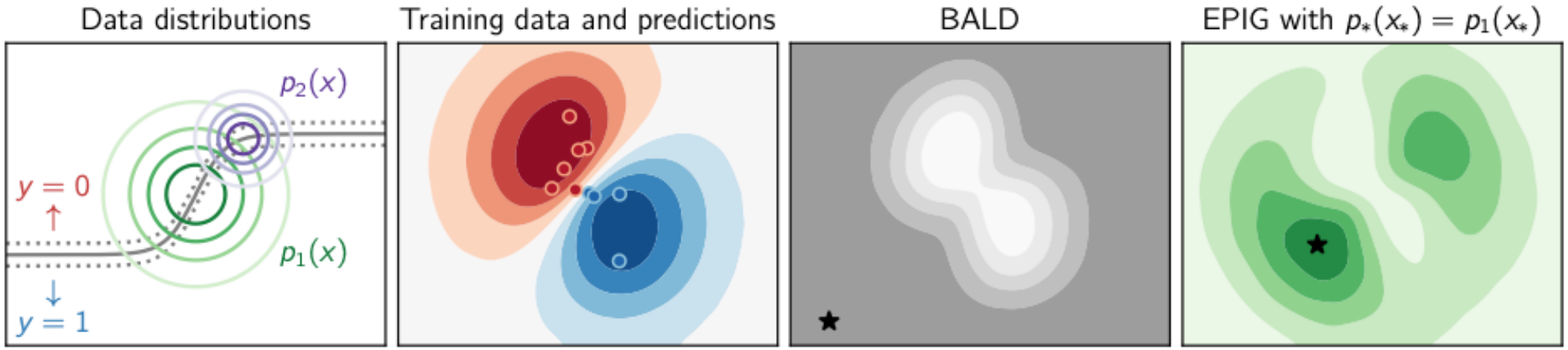
Figure 2: An illustration of when uncertainty sampling can be a poor strategy for classification. Shaded polygons represent labeled instances ($\mathcal{L}$), and circles represent unlabeled instances ($\mathcal{U}$). Since A is on the decision boundary, it will be queried as the most uncertain. However, querying B is likely to result in more information about the data as a whole.

There are 2 cases to distinguish in terms of a point being "too far away" the observed data distribution & the decision boundary. Let's look at the first now and the second case next time when speaking about overconfidence

Settles & Craven, "An Analysis of Active Learning Strategies for Sequence Labeling Tasks", EMNLP 2008

Generative models to the rescue?

The strategies we've seen have no notion of an input distribution!
We have no mechanism to ensure the data is suited to the task



There are approaches (similar to information density) that try to relate to learned data distribution, in addition to pure novelty

Smith & Kirsch et al, "Prediction-Oriented Bayesian Active Learning", AISTATS 2023

Many assumptions can come into play

What if we allow access to unlabelled pools of data?

Assumption 1: we allow a **“teacher” information source/model** that we can query

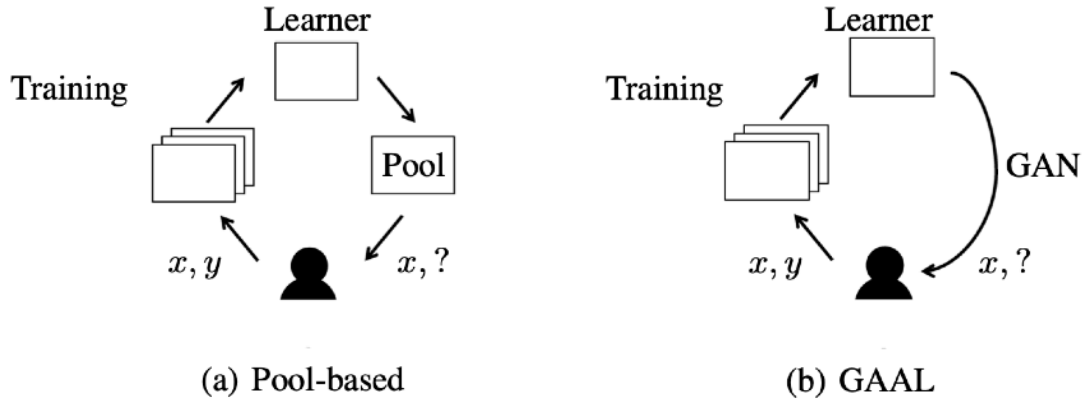
Assumption 2: we allow **semi/unsupervised training** of a generative model on an unlabelled pool

Assumption 3: we allow **clustering** or **core set** extraction mechanisms to operate on the data we wish to query

1. A “teacher” information source

Example: **generative adversarial active learning**

- Is an example of a “query-synthesizing” approach

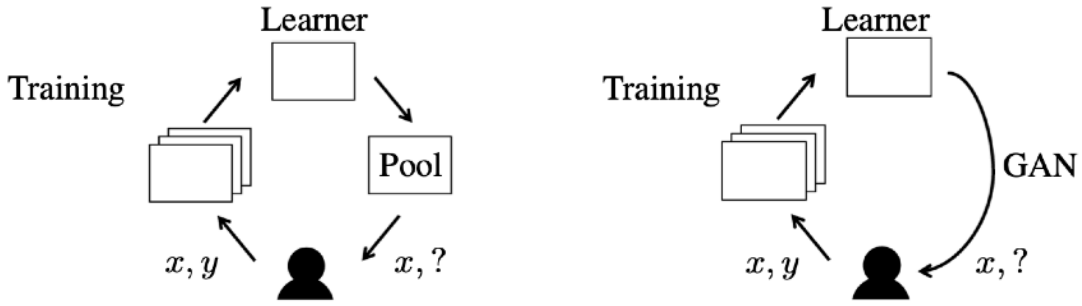


Zhu & Bento, “Generative Adversarial Active Learning”,
NeurIPS workshop on Teaching Machines, Robots & Humans, 2017

1. A “teacher” information source

Example: **generative adversarial active learning**

- Is an example of a “query-synthesizing” approach
- In essence, let generative model interpolate between known data to synthesize “novel” data with label to learn actively
- Lots of various follow-ups

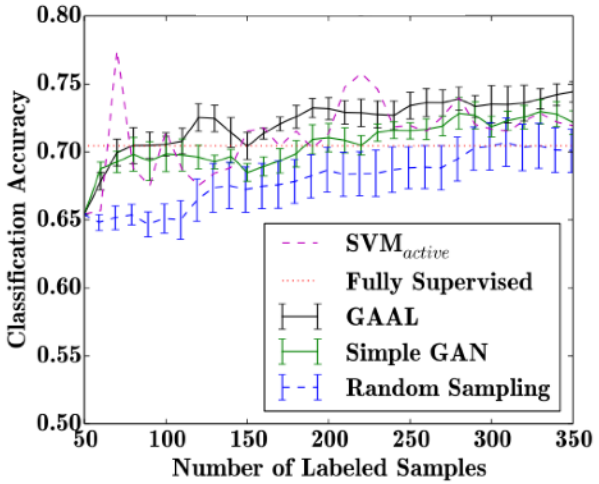


Zhu & Bento, “Generative Adversarial Active Learning”,
NeurIPS workshop on Teaching Machines, Robots & Humans, 2017

1. A “teacher” information source

Example: **generative adversarial active learning**

Figure 4: Samples generated by GAAL (Top) Generated samples in cat and dog categories. (Bottom Left) MNIST dataset. (Bottom Right) CIFAR-10 dataset.



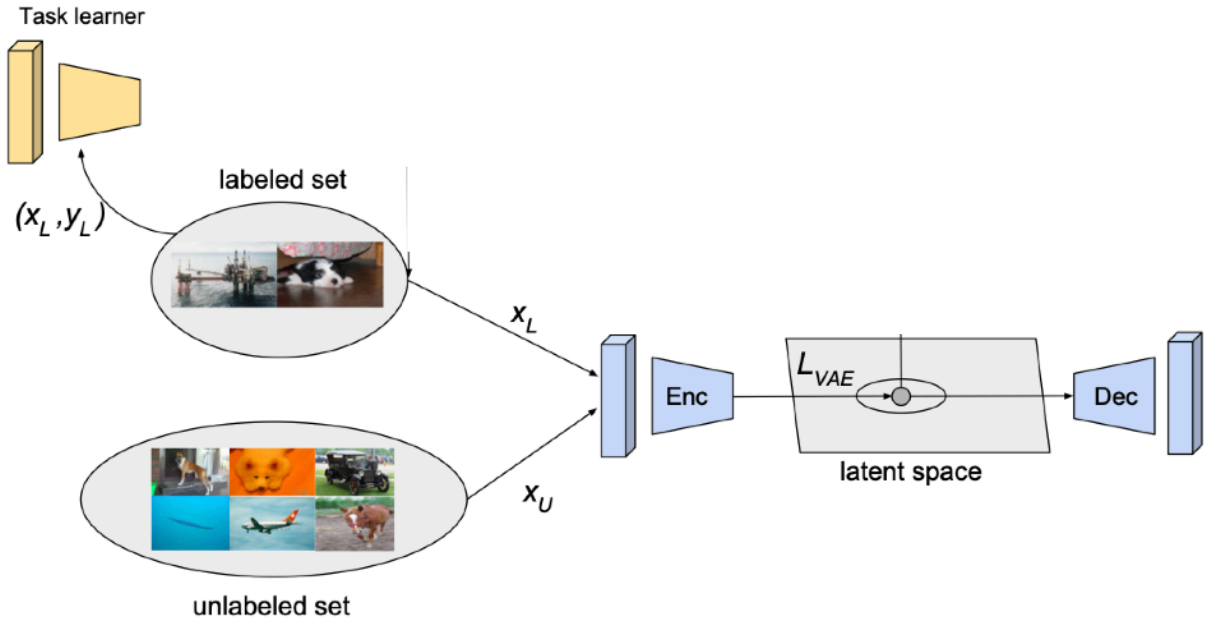
We need to be careful: how much truly “novel” instances a generative model can create (with accurate labels) - or how much a teacher model has already learned

Zhu & Bento, “Generative Adversarial Active Learning”,
NeurIPS workshop on Teaching Machines, Robots & Humans, 2017

2. Semi-supervised learning with a pool

Example: **variational (adversarial) active learning**

- Optimizes on all data
- Learns a discriminator on latent space to distinguish labelled from unlabelled

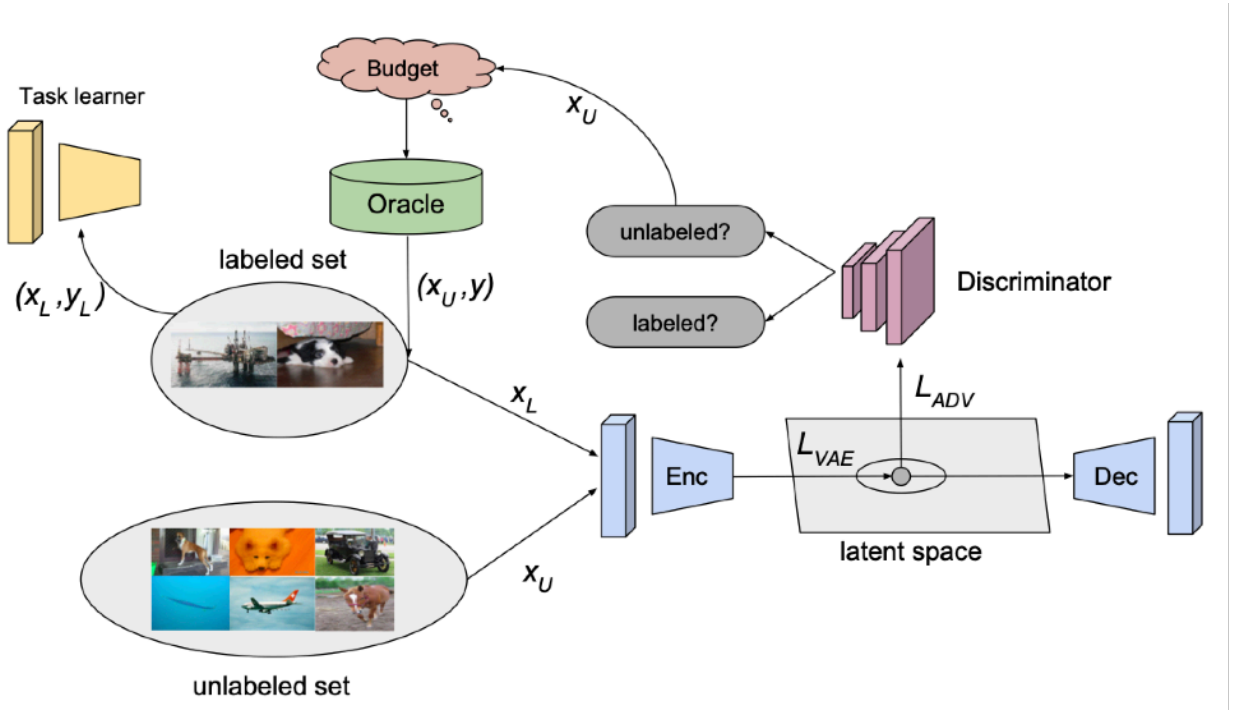


Sinha et al, “Variational Adversarial Active Learning”, ICCV 2019

2. Semi-supervised learning with a pool

Example: **variational (adversarial) active learning**

- Optimizes on all data
- Learns a discriminator on latent space to distinguish labelled from unlabelled
- Adversarial: try to fool into believing all data is labelled
- Query according to (un-)labelled confidence

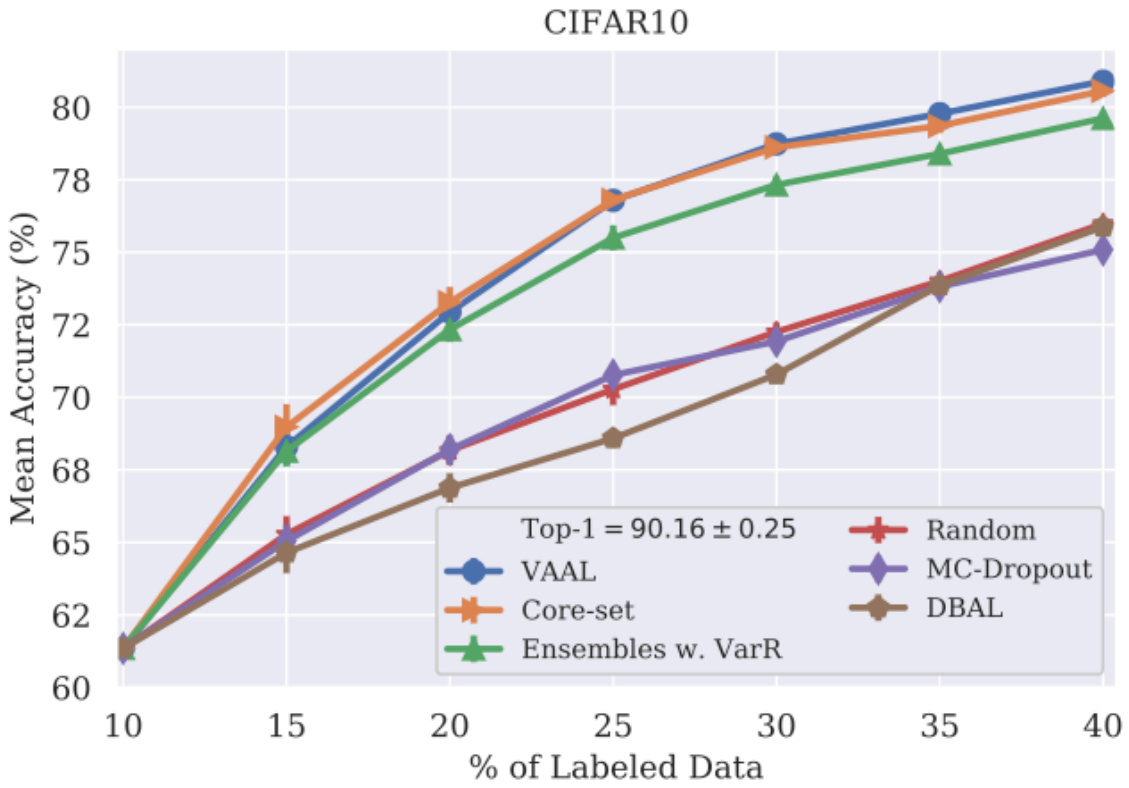


Sinha et al, “Variational Adversarial Active Learning”, ICCV 2019

2. Semi-supervised learning with a pool

Example: **variational (adversarial) active learning**

- Can be very effective



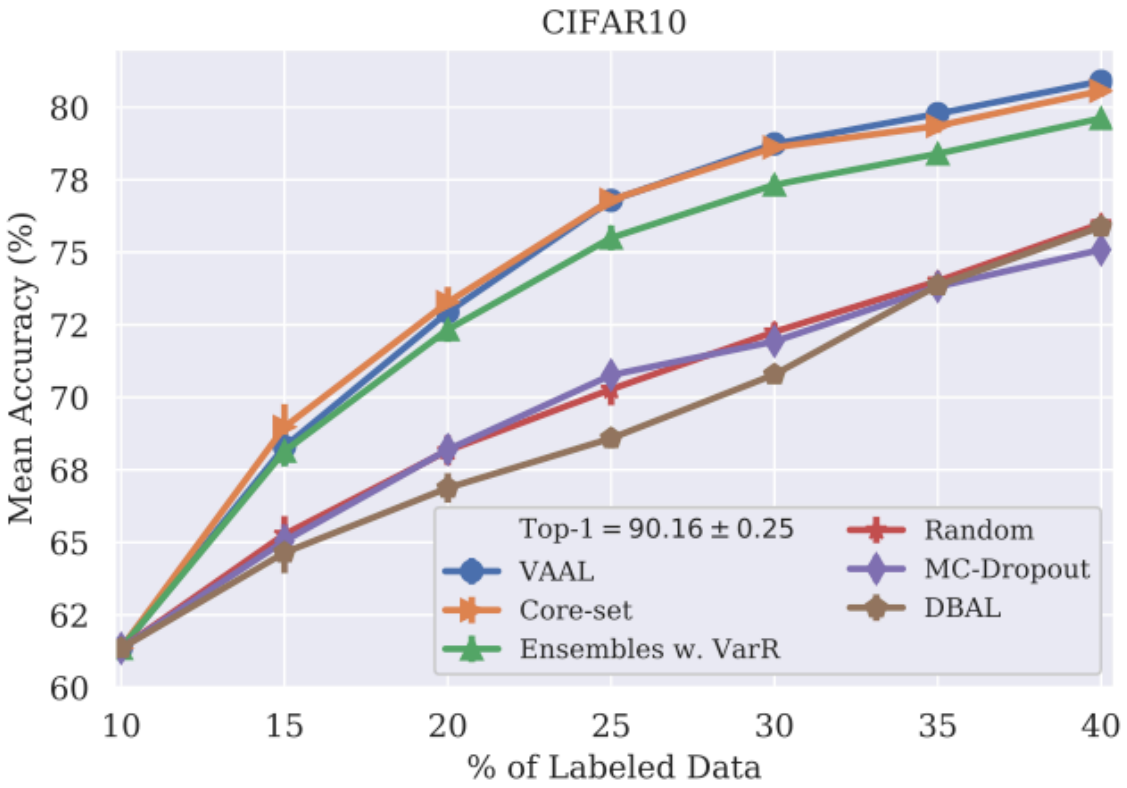
Sinha et al, “Variational Adversarial Active Learning”, ICCV 2019

2. Semi-supervised learning with a pool

Example: **variational (adversarial) active learning**

- Can be very effective

We need to be careful: pool is required up front & may be VERY large (e.g. the internet). It can contain mostly unrelated content & training can be VERY inefficient

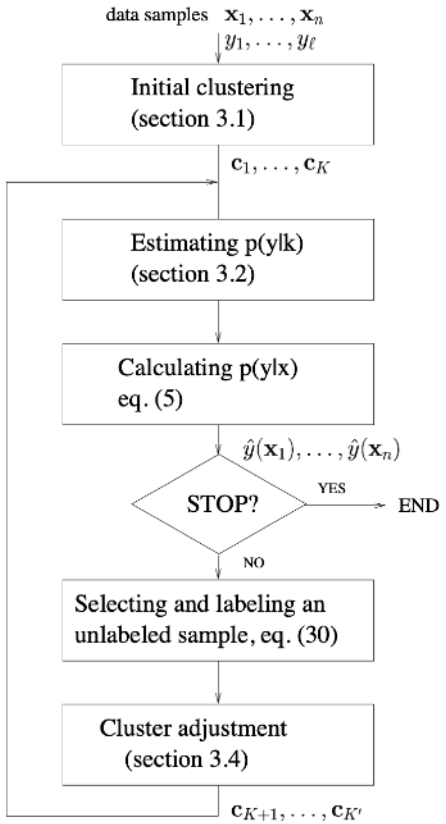


Sinha et al, “Variational Adversarial Active Learning”, ICCV 2019

3. Clusters & coresets of the pool

Example: **active Learning with pre-clustering**

- Cluster our unlabelled pool (& keep adjusting clusters over time) as the basis for query



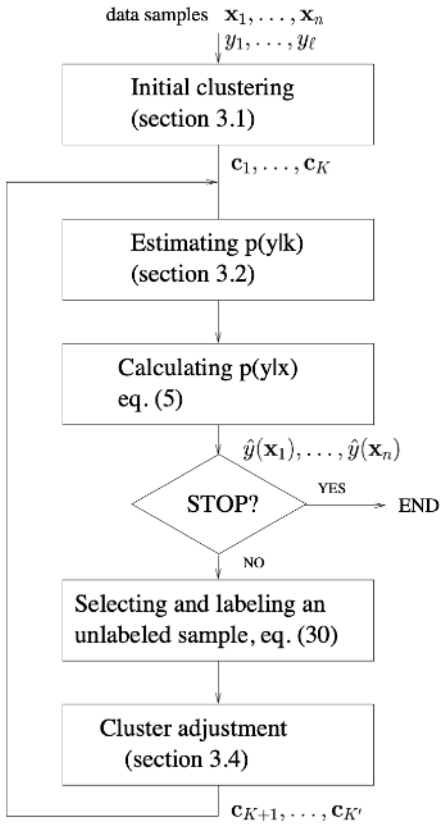
H.T. Nguyen et al, “Active Learning Using Pre-clustering”, ICML 2004

3. Clusters & coresets of the pool

Example: **active Learning with pre-clustering**

- Cluster our unlabelled pool (& keep adjusting clusters over time) as the basis for query

If we allow access to the full unlabelled pool, we might as well extract coresets



H.T. Nguyen et al, “Active Learning Using Pre-clustering”, ICML 2004

3. Clusters & coresets of the pool

Example: **coresets on the unlabelled pool**

Define core set loss as difference between loss on full set (in space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$) vs labelled subset (s) for some learning algorithm A

$$E_{\mathbf{x},y\sim p_{\mathcal{Z}}}[l(\mathbf{x},y;A_{\mathbf{s}})] \leq \underbrace{\left|E_{\mathbf{x},y\sim p_{\mathcal{Z}}}[l(\mathbf{x},y;A_{\mathbf{s}})] - \frac{1}{n}\sum_{i\in[n]} l(\mathbf{x}_i,y_i;A_{\mathbf{s}})\right|}_{\text{Generalization Error}} + \underbrace{\frac{1}{|\mathbf{s}|}\sum_{j\in\mathbf{s}} l(\mathbf{x}_j,y_j;A_{\mathbf{s}})}_{\text{Training Error}}$$

$$+ \underbrace{\left|\frac{1}{n}\sum_{i\in[n]} l(\mathbf{x}_i,y_i;A_{\mathbf{s}}) - \frac{1}{|\mathbf{s}|}\sum_{j\in\mathbf{s}} l(\mathbf{x}_j,y_j;A_{\mathbf{s}}),\right|}_{\text{Core-Set Loss}}$$

Sener & Savarese, Active Learning for CNNs: A Core-set Approach, ICLR 2018

3. Clusters & coresets of the pool

We can thus re-define the active learning problem:

"Informally, given the initial labelled set s^0 and budget (b), we are trying to find a set of points to query labels (s^1) such that when we learn a model, the performance of the model on the labelled subset and that on the whole dataset will be as close as possible"

$$\min_{\mathbf{s}^1:|\mathbf{s}^1|\leq b} \left| \frac{1}{n} \sum_{i\in[n]} l(\mathbf{x}_i, y_i; A_{\mathbf{s}^0\cup\mathbf{s}^1}) - \frac{1}{|\mathbf{s}^0 + \mathbf{s}^1|} \sum_{j\in\mathbf{s}^0\cup\mathbf{s}^1} l(\mathbf{x}_j, y_j; A_{\mathbf{s}^0\cup\mathbf{s}^1}) \right|$$

Sener & Savarese, Active Learning for CNNs: A Core-set Approach, ICLR 2018

3. Clusters & coresets of the pool

- One approach *greedy k-center:* choose b center points such that the largest distance between a data point and its nearest center is minimized.

Algorithm 1 k-Center-Greedy

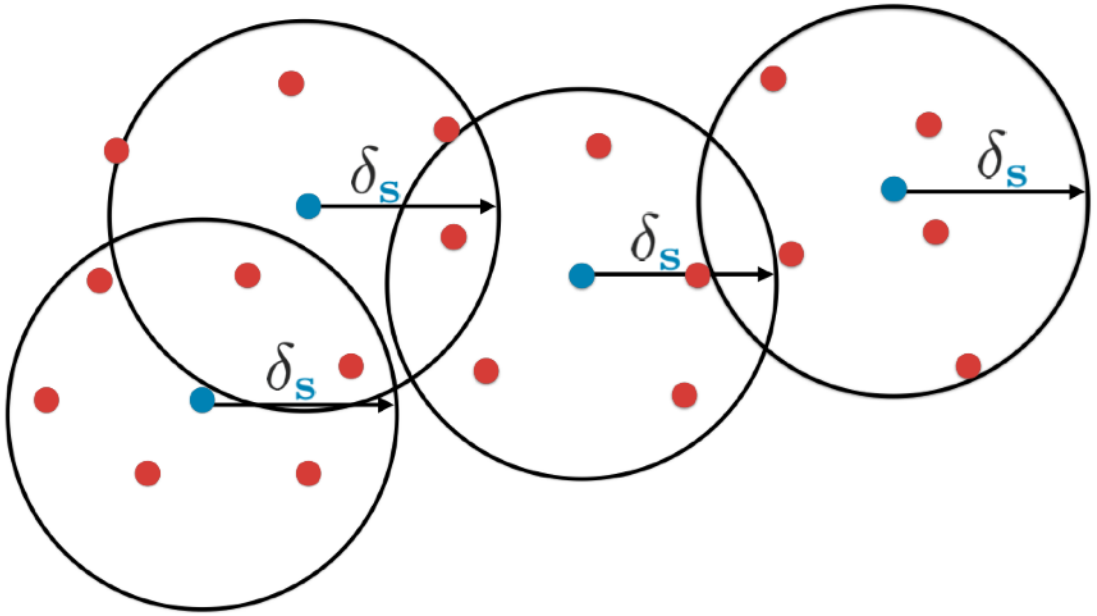
Input: data $\mathbf{x}_i$, existing pool $\mathbf{s}^0$ and a budget b
Initialize $\mathbf{s} = \mathbf{s}^0$
repeat
  $u = \arg\max_{i \in [n] \setminus \mathbf{s}} \min_{j \in \mathbf{s}} \Delta(\mathbf{x}_i, \mathbf{x}_j)$
  $\mathbf{s} = \mathbf{s} \cup \{u\}$
until $|\mathbf{s}| = b + |\mathbf{s}^0|$
return $\mathbf{s} \setminus \mathbf{s}^0$

Sener & Savarese, Active Learning for CNNs: A Core-set Approach, ICLR 2018

3. Clusters & coresets of the pool

- One approach *greedy k-center:* choose b center points such that the largest distance between a data point and its nearest center is minimized.

- In theory, we try to find the "set-cover": *"a set s is a δ cover of a set $s^\star$ means a set of balls with radius δ centered at each member of s can cover the entire $s^\star$"*

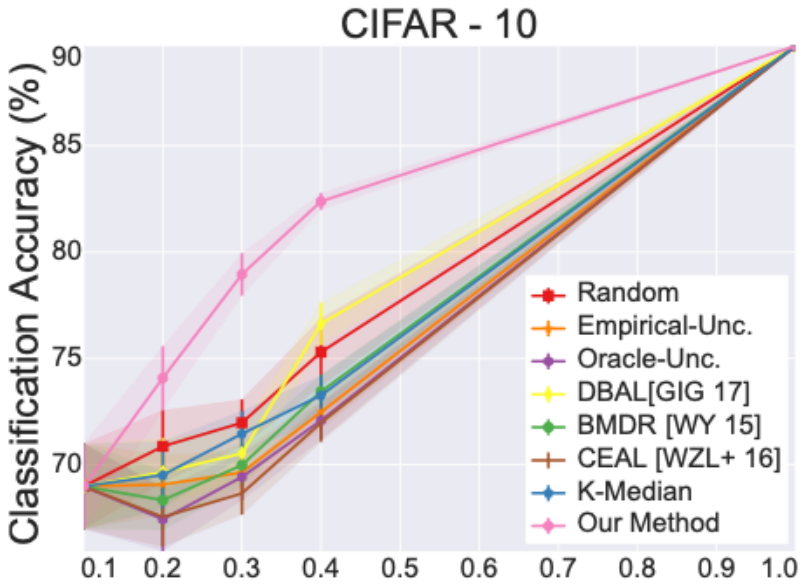


Sener & Savarese, Active Learning for CNNs: A Core-set Approach, ICLR 2018

3. Clusters & coresets of the pool

- One approach *greedy k-center:* choose b center points such that the largest distance between a data point and its nearest center is minimized.

- In theory, we try to find the "set-cover": *"a set s is a δ cover of a set $s^\star$ means a set of balls with radius δ centered at each member of s can cover the entire $s^\star$"*

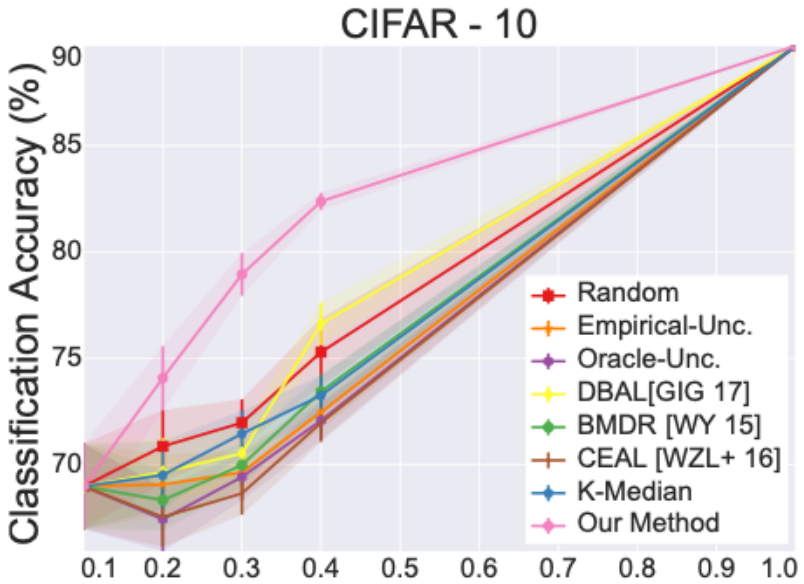


Sener & Savarese, Active Learning for CNNs: A Core-set Approach, ICLR 2018

3. Clusters & coresets of the pool

- One approach *greedy k-center:* choose b center points such that the largest distance between a data point and its nearest center is minimized.

- In theory, we try to find the "set-cover": *"a set s is a δ cover of a set $s^\star$ means a set of balls with radius δ centered at each member of s can cover the entire $s^\star$"*



We will learn about covers in more detail as it's a VERY powerful approach, but first let us learn more about what is holding us back

Sener & Savarese, Active Learning for CNNs: A Core-set Approach, ICLR 2018

Universität Bremen

OWL ML

Question time

We had different assumptions & trade-offs so far. Can you summarize them?

Acquisition function perspectives

Version space reduction -> Hypothesis

The formal approach: reduce the set/space of possible hypotheses: $h : \mathcal{X} \rightarrow \mathcal{Y}$ by removing ones that are inconsistent with the data

Uncertainty & heuristics -> Novelty

The intuitive approach: use the predictions, or maybe even better, uncertainty in the predictions for the queries

Representation learning (& coresets) -> Coverage/“Diversity”

The distribution based approach: maximizing coverage instead of reducing the possible set of hypotheses (version space) explicitly

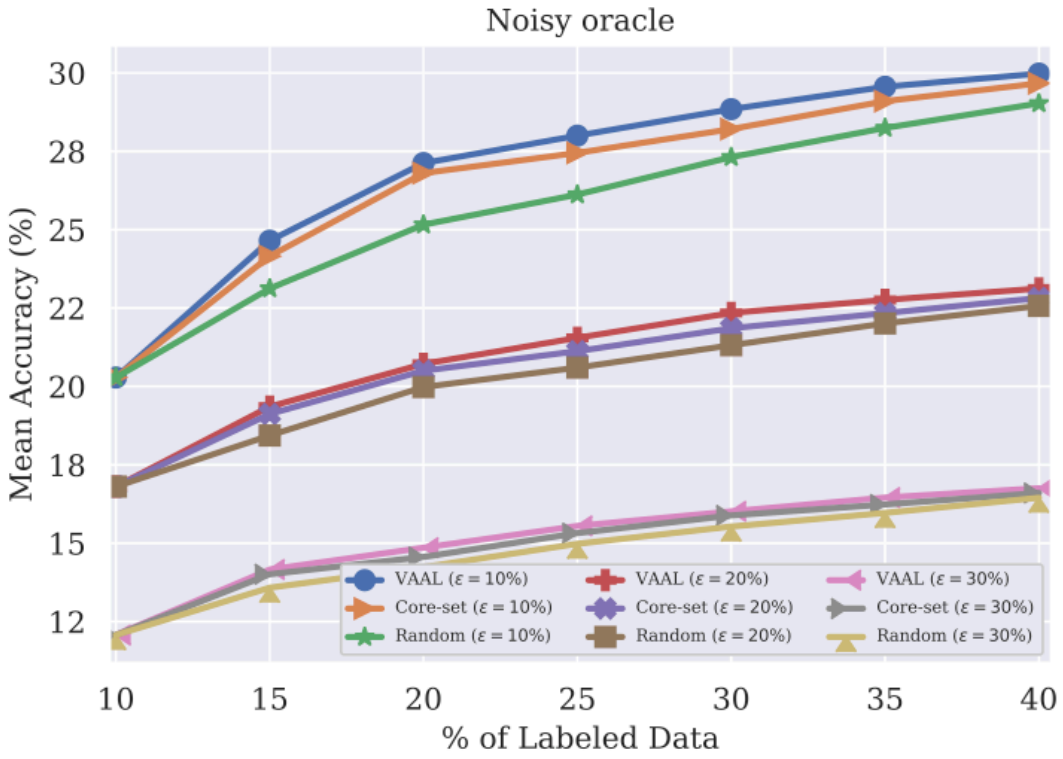
Active learning strategies in summary

Techniques	& (some of) their assumptions
• Version space reduction	• Set of hypotheses is clear
• Minimum confidence • Maximum entropy • Best versus second best	• No overconfidence phenomenon and out-of-distribution/task data
• Model “uncertainty” (output variability) • Ensembles/query by committee	• Accurate uncertainty everywhere • Training of multiple models
• Representation learning on the pool • Core sets	• Upfront training on entire pool (access + computational expense)

Acquisition function perspectives

The big assumptions that cause massive failure when not valid

- *Oracle is infallible*:
  no teacher/labeler mistakes!
- *Data is accumulated*:
  no “continual” active learning
- *Pool belongs to task*:
  no interfering/obsolete data

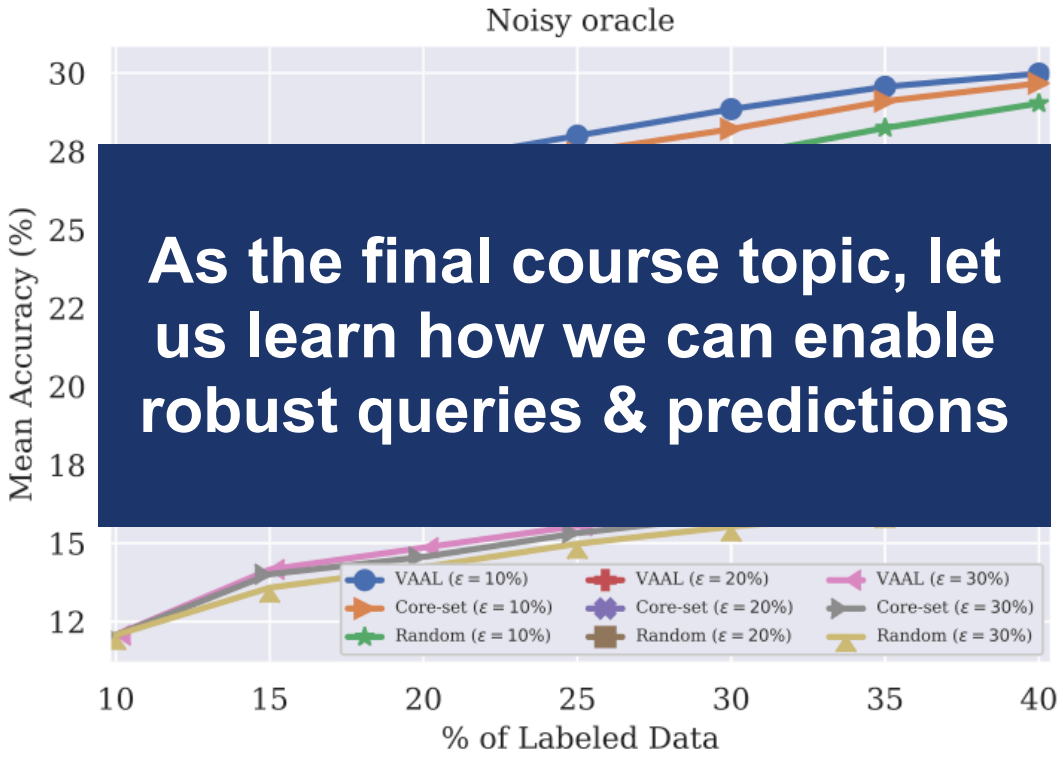


Sinha et al, “Variational Adversarial Active Learning”, ICCV 2019

Acquisition function perspectives

The big assumptions that cause massive failure when not valid

- *Oracle is infallible*:
  no teacher/labeler mistakes!
- *Data is accumulated*:
  no “continual” active learning
- *Pool belongs to task*:
  no interfering/obsolete data



As the final course topic, let us learn how we can enable robust queries & predictions

Sinha et al, “Variational Adversarial Active Learning”, ICCV 2019

In the “real world” it’s more than that

We realistically encounter:

- Distribution shift & sampling bias
- Corrupted data

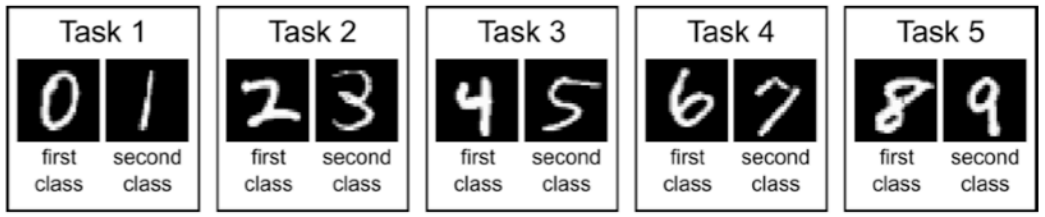
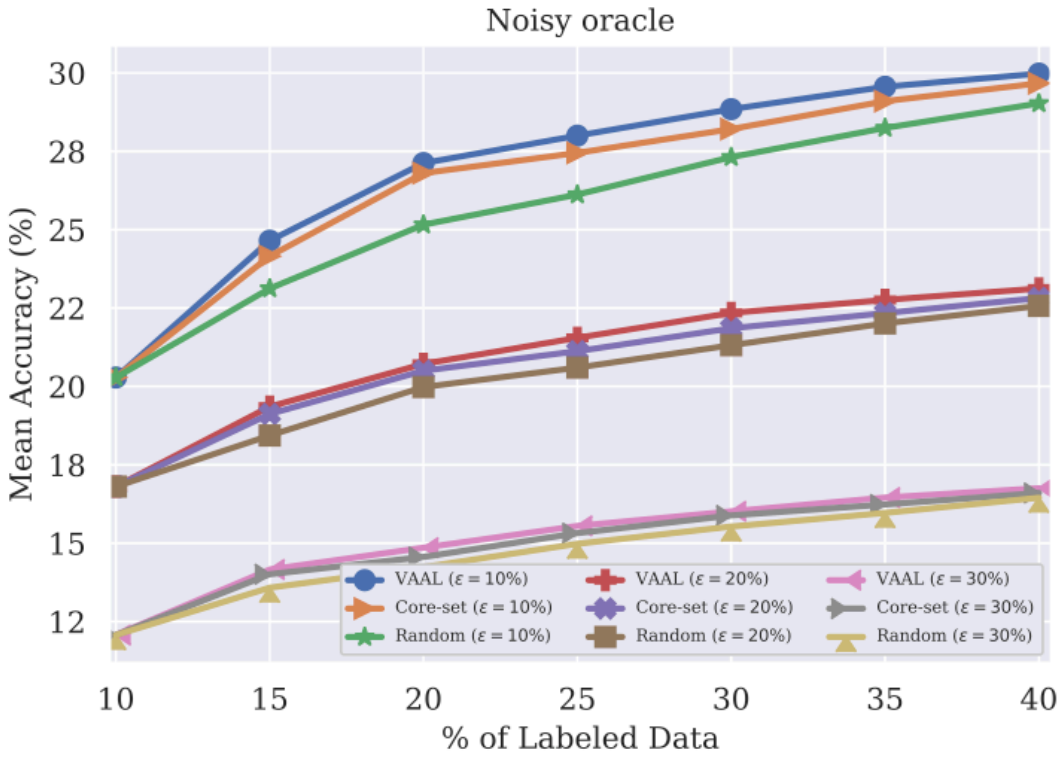


Figure 1: Schematic of split MNIST task protocol.

van de Ven et al, “Three types of incremental learning”, Nature MI 2022



Sinha et al, “Variational Adversarial Active Learning”, ICCV 2019

In the “real world” it’s more than that

We realistically encounter:

- Distribution shift & sampling bias
- Corrupted data
- Blurred task boundaries
- Unrelated & uninformative data

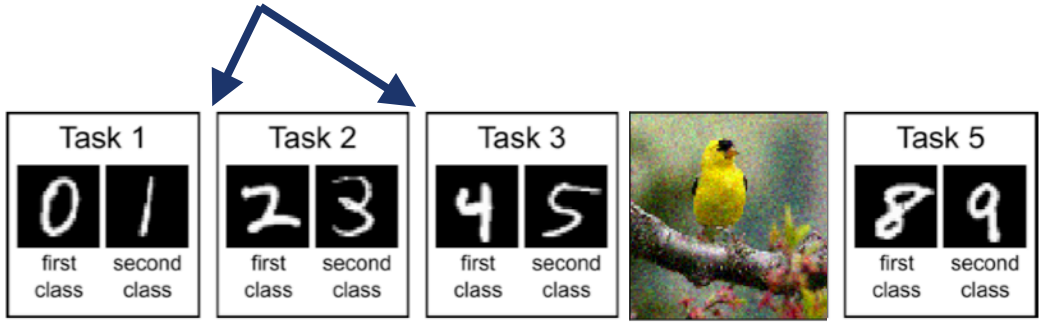
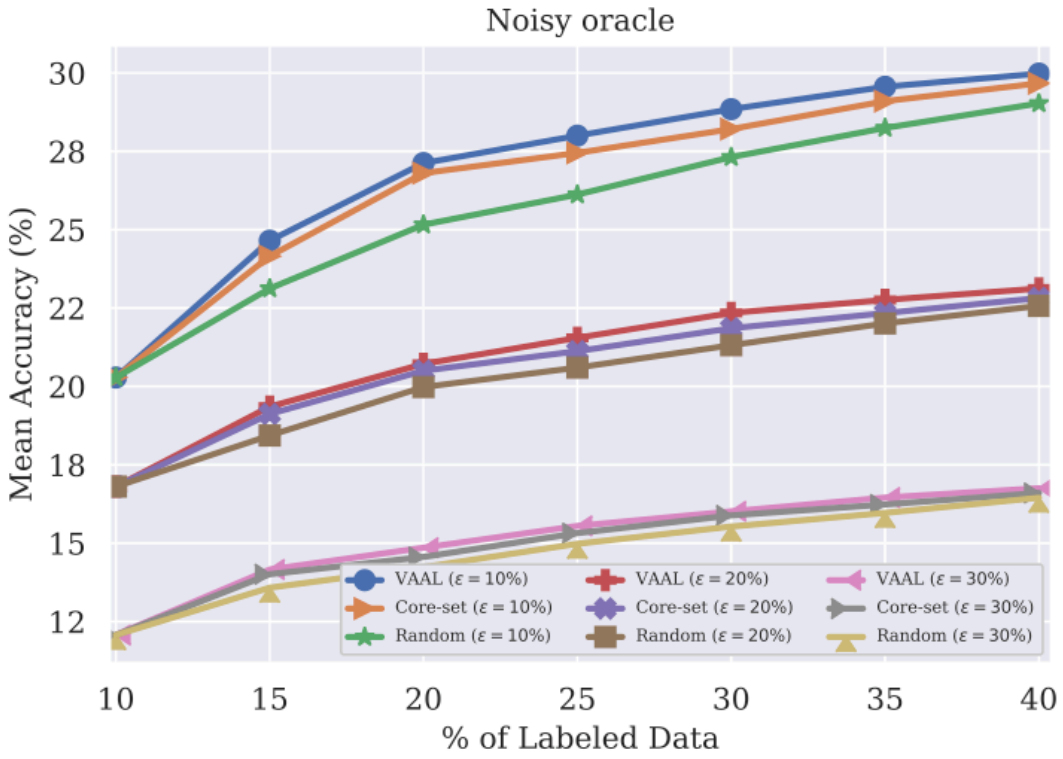


Figure 1: Schematic of split MNIST task protocol.

van de Ven et al, “Three types of incremental learning”, Nature MI 2022



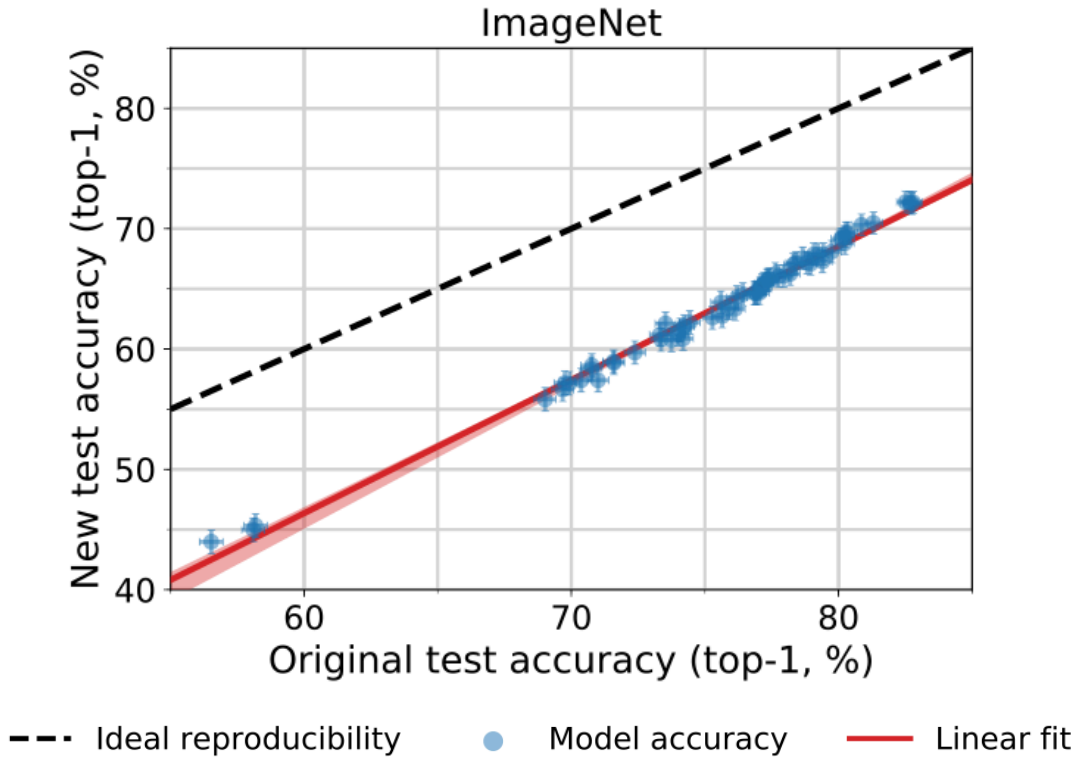
Sinha et al, “Variational Adversarial Active Learning”, ICCV 2019

A real world motivational example: ImageNet

An ImageNet Experiment:

- Recollect a second test set following the exact same instructions of the original dataset acquisition procedure

Every single model developed on ImageNet (in the study) performed consistently worse - it did not really "generalize"



Recht et al, "Do ImageNet Classifiers Generalize to ImageNet?", ICML 2019

Universität Bremen

OWL ML

Question time

Earlier findings & assumptions prompt questions:
What is “known” and what is “unknown”?
Can you think of different cases of what we should and shouldn’t be able to know?

Universität Bremen

More than “known” versus “unknown”

Known knowns:
Do you have any intuition what these 4 categories could represent?

Known unknowns:

Unknown unknowns:

Unknown knowns:

More than “known” versus “unknown”

Known knowns:
Examples belong to the distribution from training set was drawn.
Assumption of an accurate & confident prediction.

Known unknowns:

Unknown unknowns:

Unknown knowns:

Universität Bremen

More than “known” versus “unknown”

Known knowns:
Examples belong to the distribution from training set was drawn.
Assumption of an accurate & confident prediction.

Known unknowns:
Unknown examples where models have high uncertainty. Can be optionally “negatively” labelled examples used in training.

Unknown unknowns:

Unknown knowns:

More than “known” versus “unknown”

Known knowns:
Examples belong to the distribution from training set was drawn.
Assumption of an accurate & confident prediction.

Known unknowns:
Unknown examples where models have high uncertainty. Can be optionally “negatively” labelled examples used in training.

Unknown unknowns:
Unseen instances belonging to unexplored & unknown distributions.
Predictions generally overconfident & by definition false.

Unknown knowns:

Universität Bremen

More than “known” versus “unknown”

Known knowns:
Examples belong to the distribution from training set was drawn.
Assumption of an accurate & confident prediction.

Known unknowns:
Unknown examples where models have high uncertainty. Can be optionally “negatively” labelled examples used in training.

Unknown unknowns:
Unseen instances belonging to unexplored & unknown distributions.
Predictions generally overconfident & by definition false.

Unknown knowns:
We know the concept but choose to treat it as unknown. Can be either willful ignorance or e.g. filtering/masking controversial content

More than “known” versus “unknown”

Known knowns (or simply knowns):
Examples belong to the distribution from training set was drawn.
Assumption of an accurate & confident prediction.

Known unknowns:
Unknown examples where models have high uncertainty. Can be optionally “negatively” labelled examples used in training.

Unknown unknowns:
Unseen instances belonging to unexplored & unknown distributions.
Predictions generally overconfident & by definition false.

Unknown knowns (often not considered):
We know the concept but choose to treat it as unknown. Can be either willful ignorance or e.g. filtering/masking controversial content

Question time

How can we recognize these separate cases?

Universität Bremen

Three types of approaches

Prediction anomalies:
Out-of-distribution are hopefully separable through anomalous outputs

Universität Bremen

Three types of approaches

Prediction anomalies:
Out-of-distribution are hopefully separable through anomalous outputs

Before we explore the other two type of approaches, let us learn about the intuitive, but very unfortunate part of the story first

Disclaimer: I will use figures from our own publications for convenience, but the observed patterns are MUCH older (1990s, e.g. Matan et al)

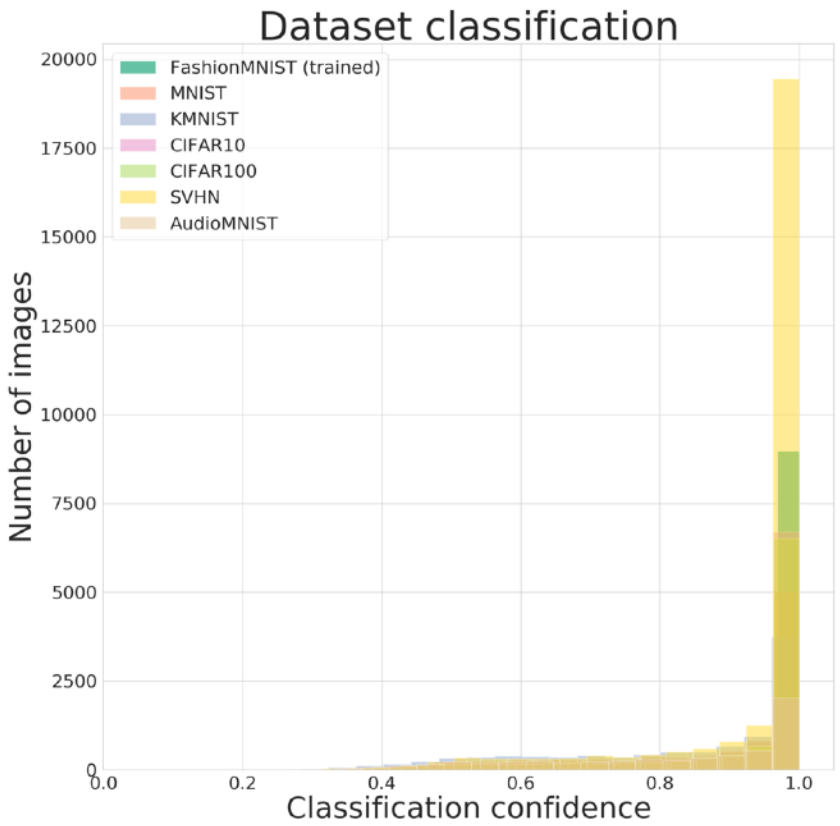
Universität Bremen

The intuitive unfortunate part: overconfidence

Consider a quantitative example:

1. Train a (neural network) classifier on a dataset (here fashion items)
2. Log predictions on the test set (here green)
3. Log predictions on arbitrary other data (here other colors)

The intuitive unfortunate part: overconfidence

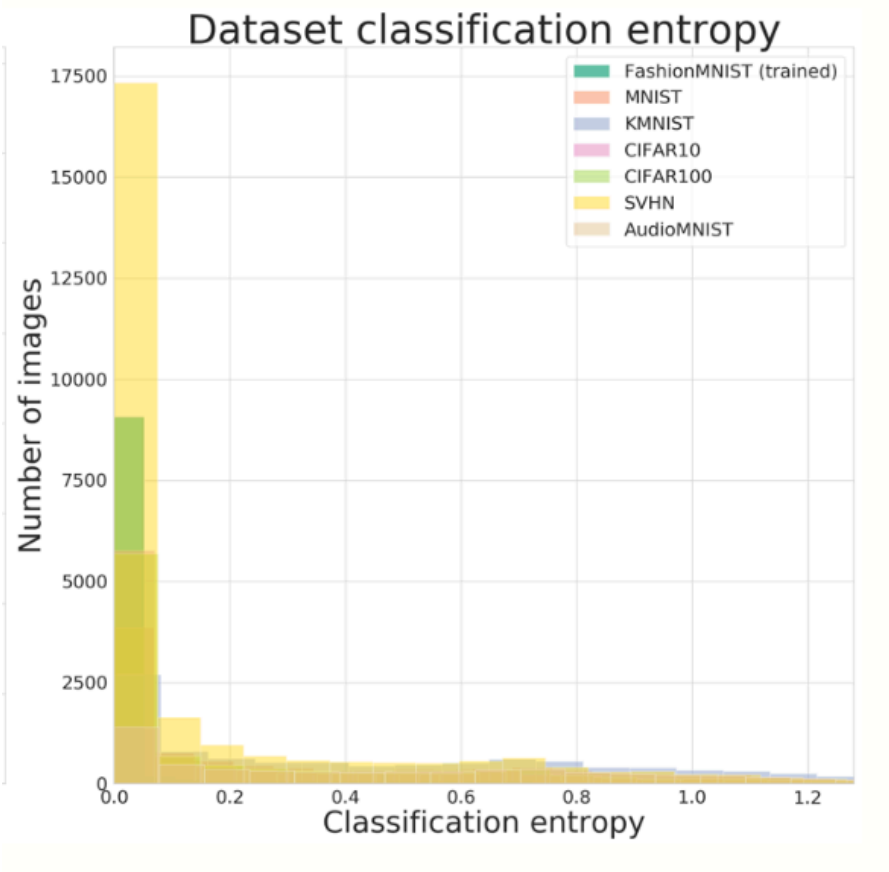


Consider a quantitative example:

1. Train a (neural network) classifier on a dataset (here fashion items)
2. Log predictions on the test set (here green)
3. Log predictions on arbitrary other data (here other colors)
4. Observe that majority of misclassifications happen with large output confidence/probability

Figure: Mundt et al “Open Set Recognition Through Deep Neural Network Uncertainty”, ICCVW 2019

The intuitive unfortunate part: overconfidence



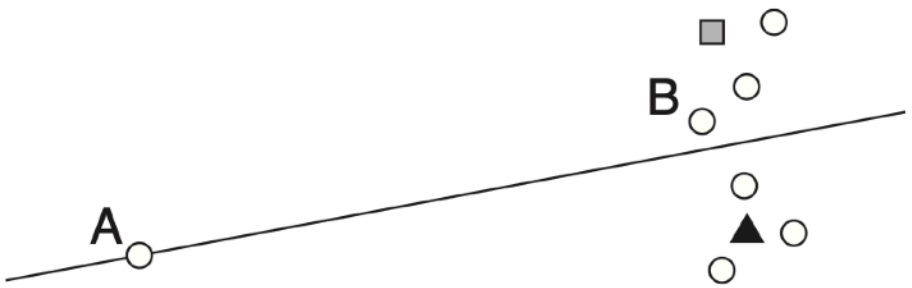
Recall that in traditional active learning, we motivated the use of entropy and other information theoretic measures, but assumed task relatedness (allowing only noise)

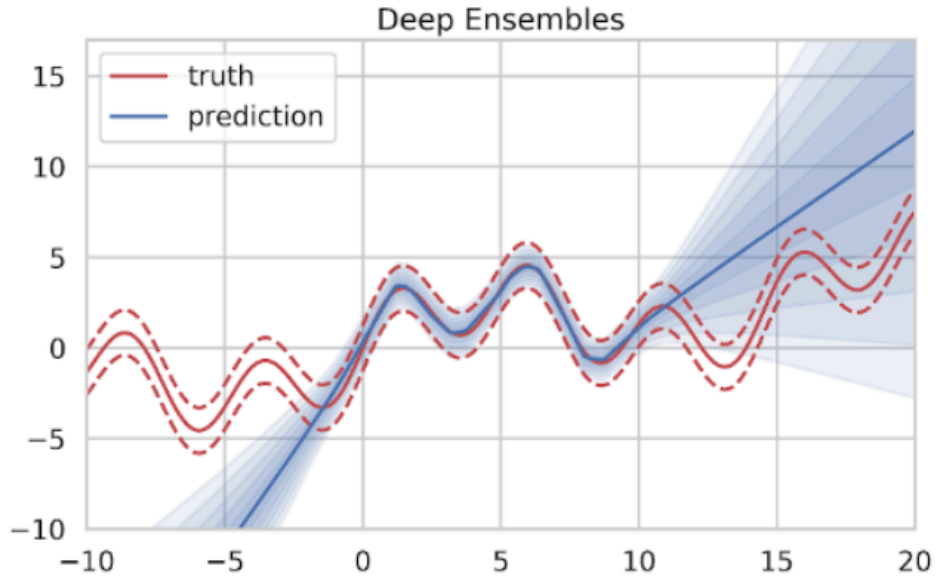
If we allow for fully unrelated task data to exist, novelty centered measures fail to distinguish relevant from irrelevant data

Figure: Mundt et al “Open Set Recognition Through Deep Neural Network Uncertainty”, ICCVW 2019

Predictive uncertainty is not an easy “fix”

Recall that in active learning we said there are some cases that model uncertainty struggles with - these are now imperative





Settles & Craven, "An Analysis of Active Learning Strategies for Sequence Labeling Tasks”, EMNLP 2008 & https://www.inovex.de/de/blog/uncertainty-quantification-deep-learning/

Predictive uncertainty is not an easy “fix”

Let’s look at how Monte-Carlo Dropout improves separation

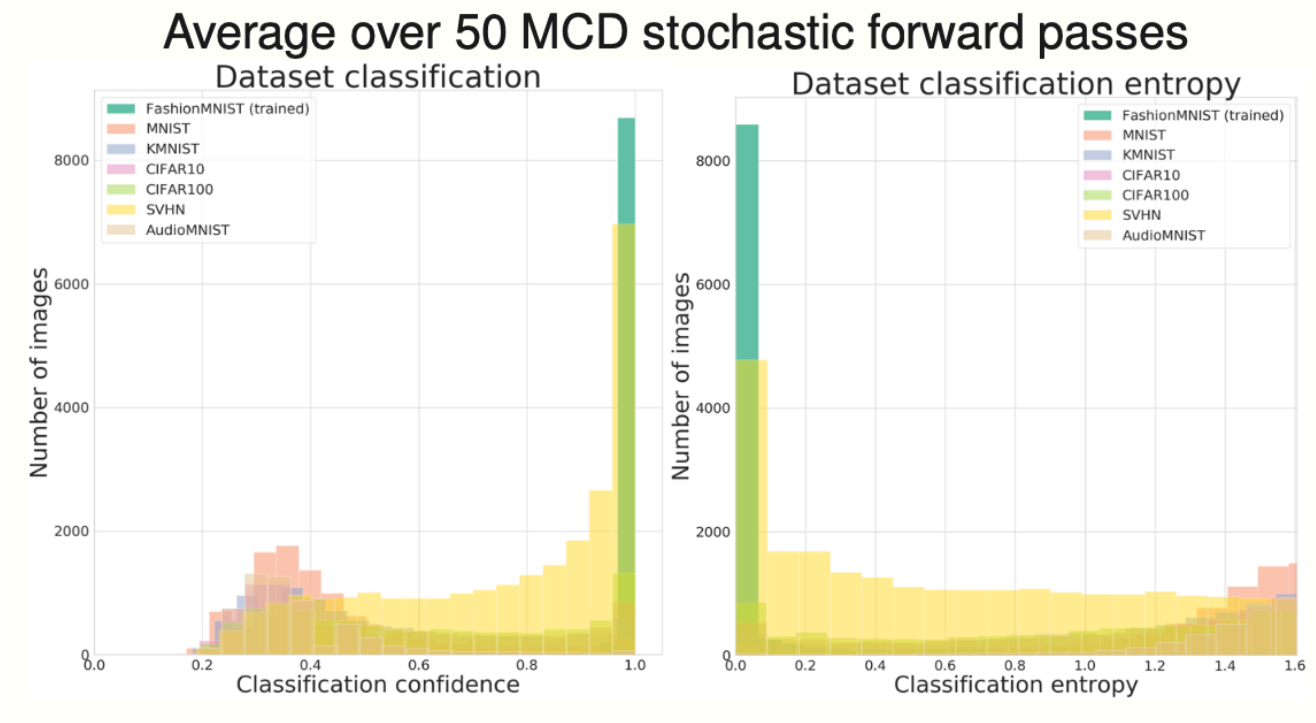
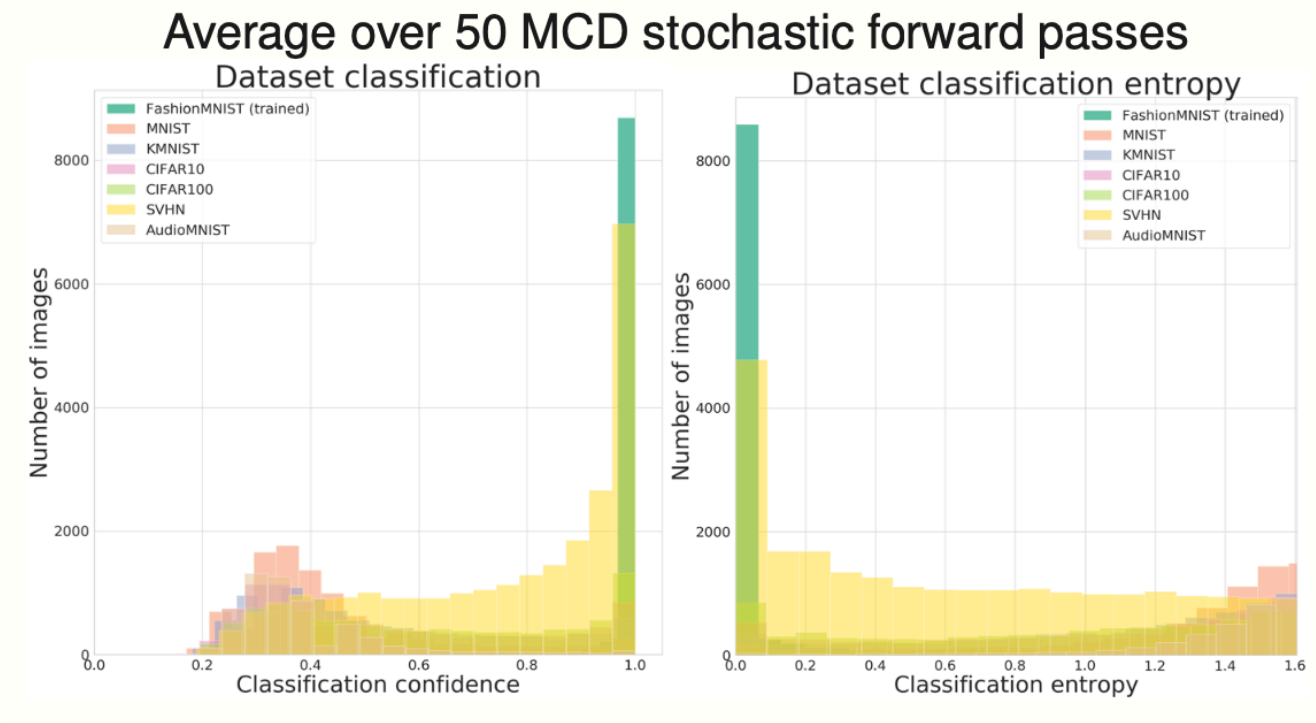


Figure: Mundt et al “Open Set Recognition Through Deep Neural Network Uncertainty”, ICCVW 2019

Predictive uncertainty is not an easy “fix”

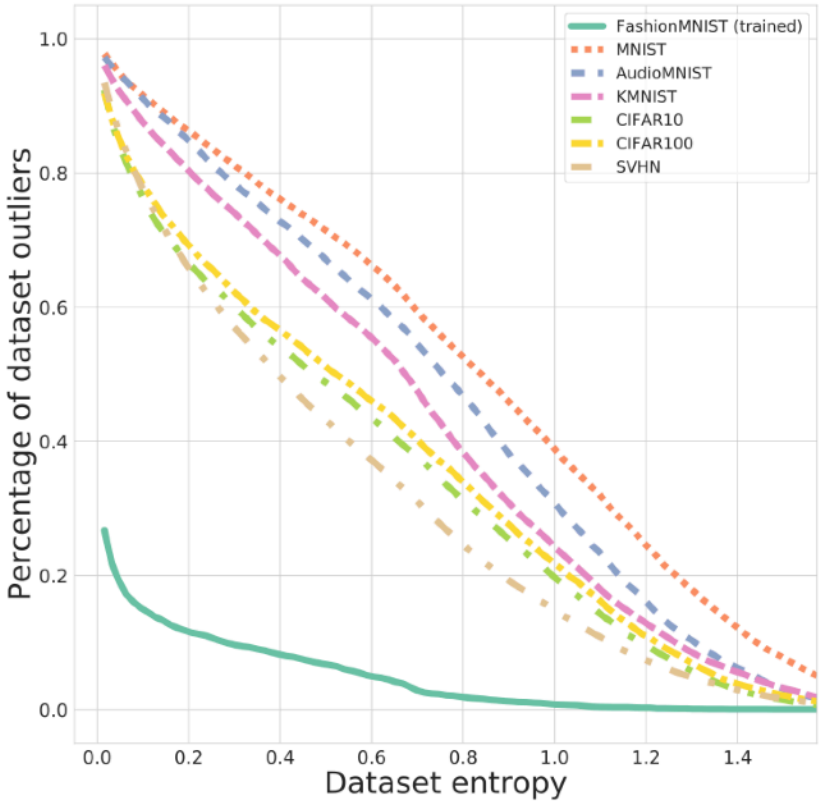
Let’s look at how Monte-Carlo Dropout improves separation



It doesn’t look too bad at first glance - but there is still substantial overlap & importantly: we need to find a threshold!

Figure: Mundt et al “Open Set Recognition Through Deep Neural Network Uncertainty”, ICCVW 2019

Predictive anomalies & thresholds

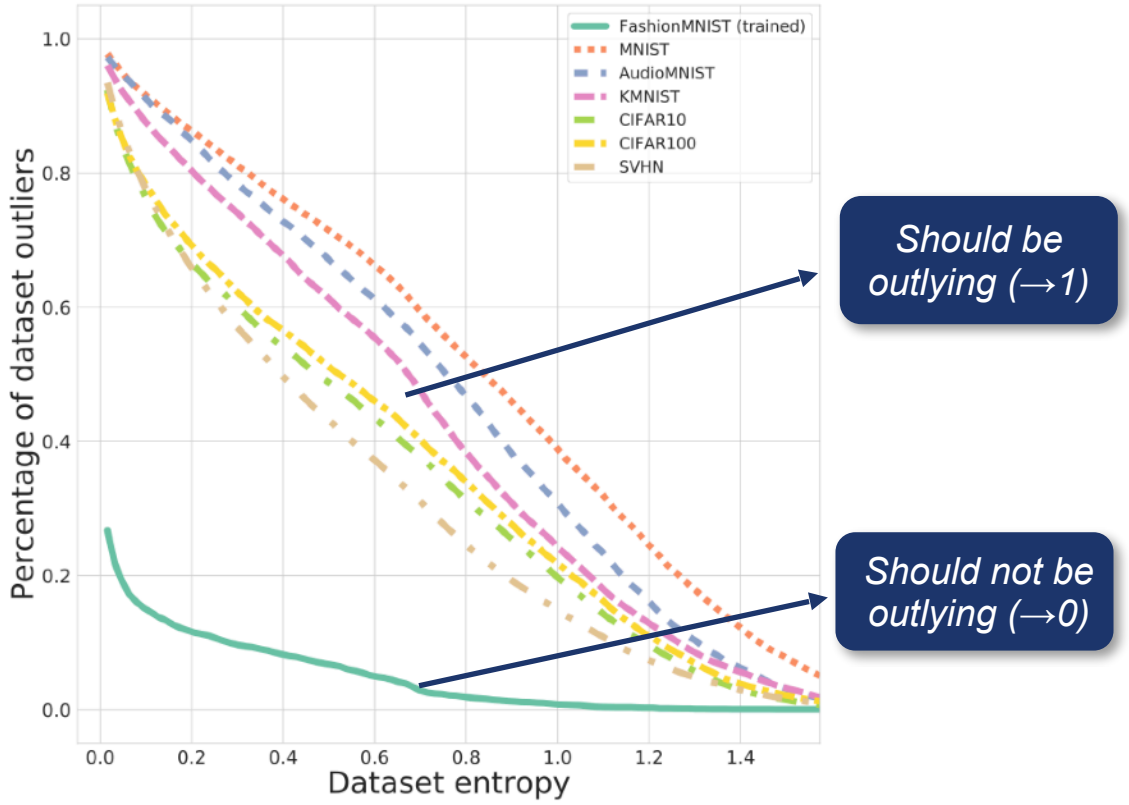


We will have to pick a threshold on a validation set

What should these curves ideally look like?

Figure: Mundt et al, “Unified Probabilistic Deep Continual Learning Through Open Set Recognition and Generative Replay”, Journal of Imaging, Volume 8, Issue 4, 2022

Predictive anomalies & thresholds

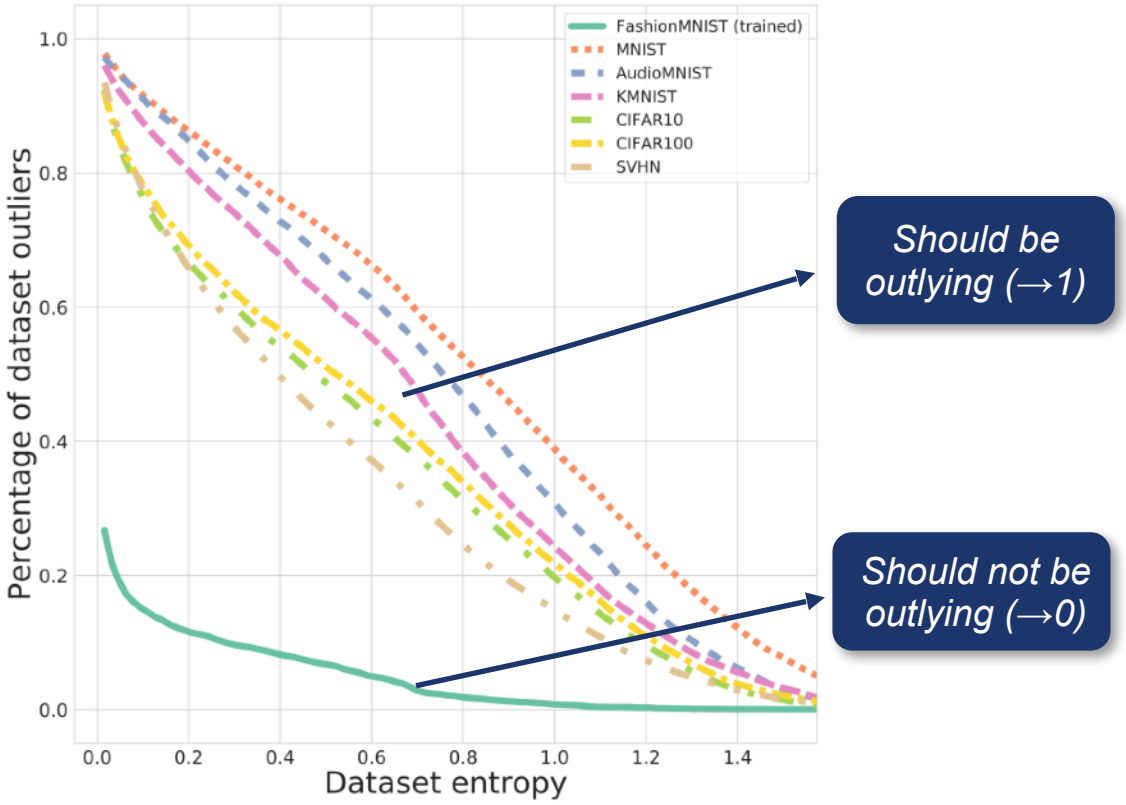


We will have to pick a threshold on a validation set

We are essentially trading off mistakes on in-domain vs. out-of-domain data (without fully succeeding)

Figure: Mundt et al, “Unified Probabilistic Deep Continual Learning Through Open Set Recognition and Generative Replay”, Journal of Imaging, Volume 8, Issue 4, 2022

Predictive anomalies & thresholds



We will have to pick a threshold on a validation set

We are essentially trading off mistakes on in-domain vs. out-of-domain data (without fully succeeding)

Would we succeed if we picked a generative model instead?

Figure: Mundt et al, “Unified Probabilistic Deep Continual Learning Through Open Set Recognition and Generative Replay”, Journal of Imaging, Volume 8, Issue 4, 2022

Predictive anomalies & thresholds

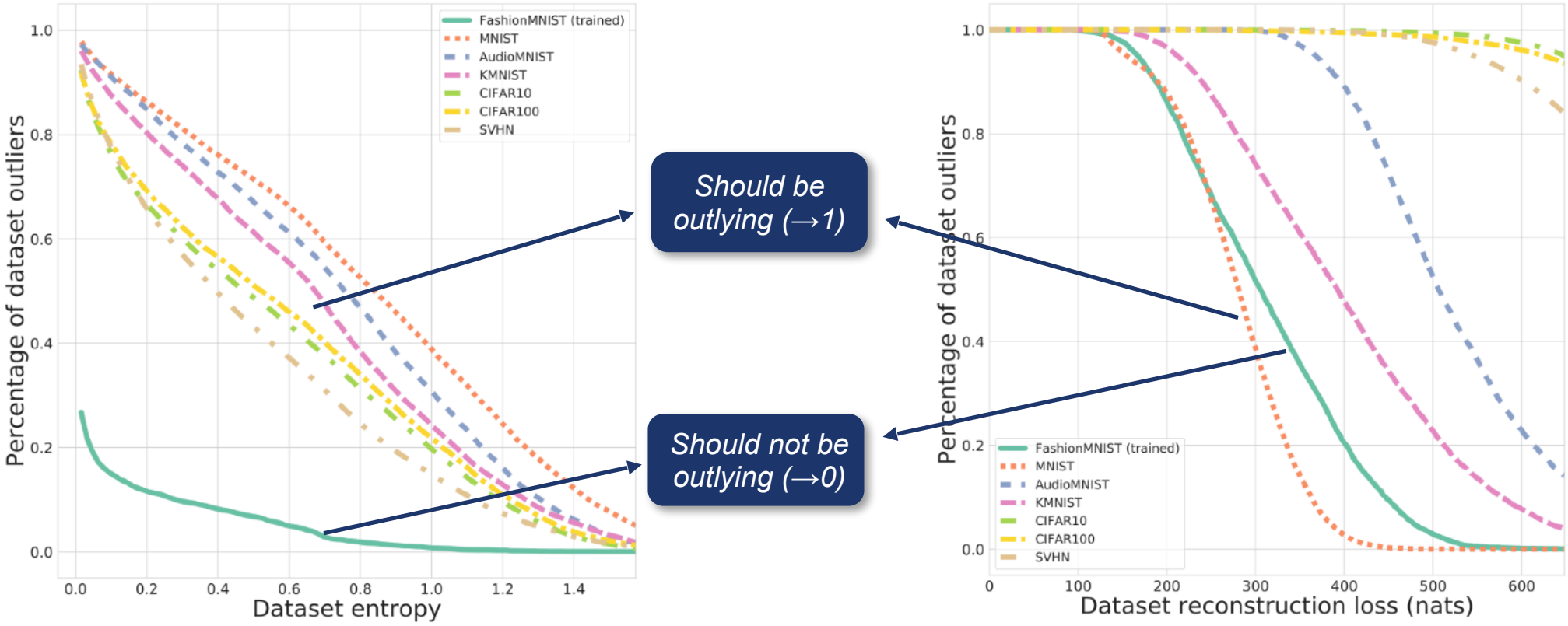
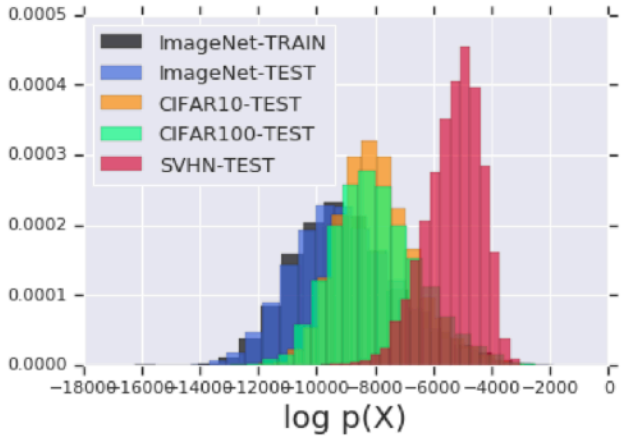


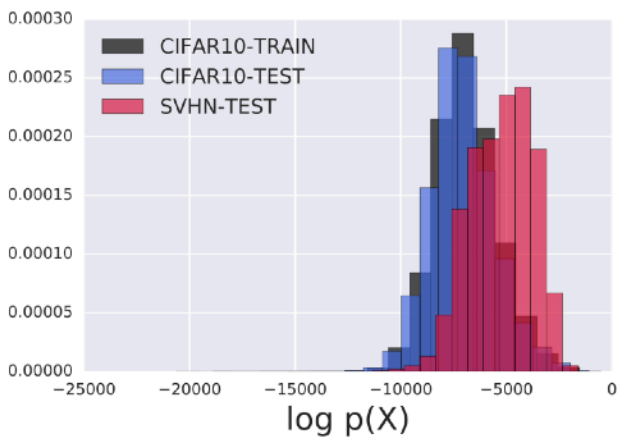
Figure: Mundt et al, “Unified Probabilistic Deep Continual Learning Through Open Set Recognition and Generative Replay”, Journal of Imaging, Volume 8, Issue 4, 2022

Overconfidence in generative models

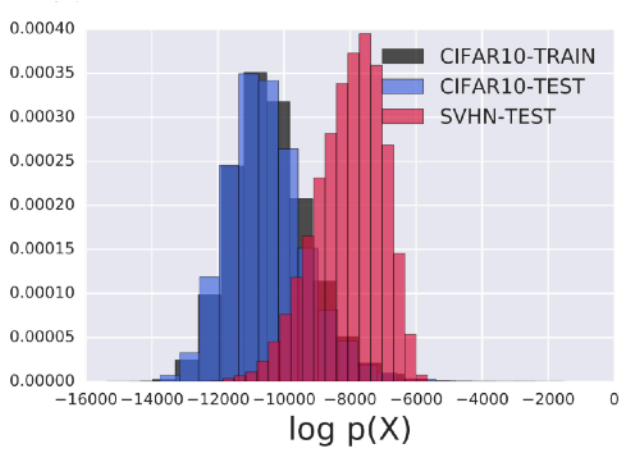
This overconfidence pattern is observable for many gen. models



Glow
(Normalizing Flow)



PixelCNN
(Autoregression)



VAE
(Variational Inference)

Nalisnick et al, “Do Deep Generative Models Know What They Don’t Know”, ICLR 2019

Universität Bremen

OWL ML

Question time

Why do we systematically observe overconfidence?
Can you think of technical reasons?

Predictive anomalies & thresholds

Multiple reasons, but some are:

1. We train only to maximize probabilities / minimize loss:

$$R_{emp}(D, \delta) = 1/N \sum_{i=1}^{N} L(y_i, \delta(x_i))$$

Predictive anomalies & thresholds

Multiple reasons, but some are:

1. We train only to maximize probabilities / minimize loss:

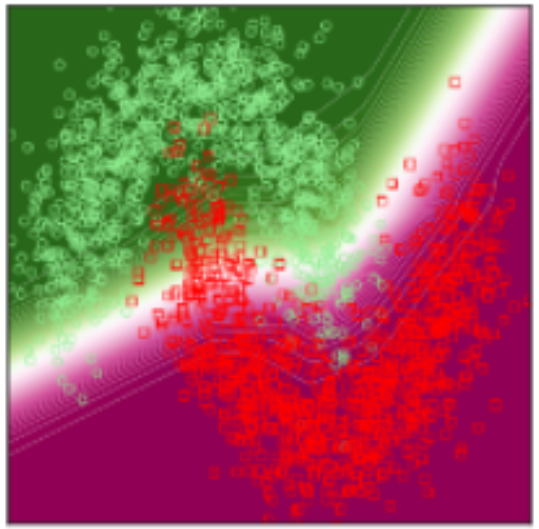
$$R_{emp}(D, \delta) = 1/N \sum_{i=1}^{N} L(y_i, \delta(x_i))$$

2. Specific function choices in ML models are unbounded

(e,g. Hein et al, “Why ReLU networks yield high confidence predictions far away from the training data and how to mitigate the problem”, CVPR 2019)

ReLU: Rectified Linear Unit

Zoomed In



Zoomed Out

Thanks to Quentin Delfosse for producing the figure on activation function overconfidence

Question time

If these are the reasons, can you think of (comparably easy) ways to fix them?

Three types of approaches

Prediction anomalies:
Out-of-distribution are hopefully separable through anomalous outputs

Incorporating prior knowledge:
Include “background” or “non-example” data population explicitly

Open set recognition:
Rely on predictions from the “covered” space; we separate closed/open sets

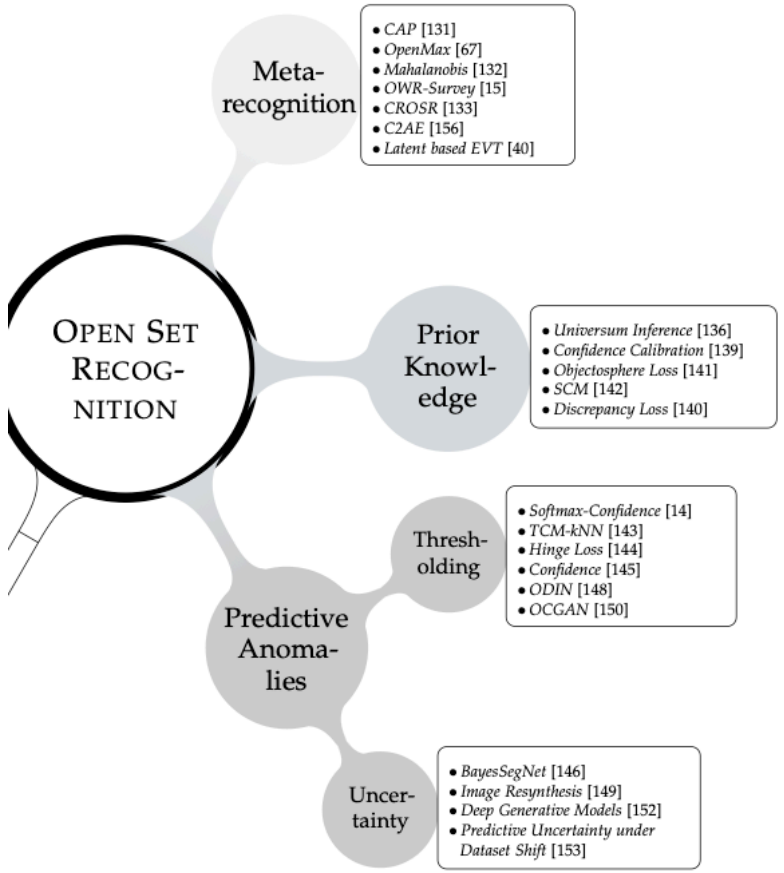
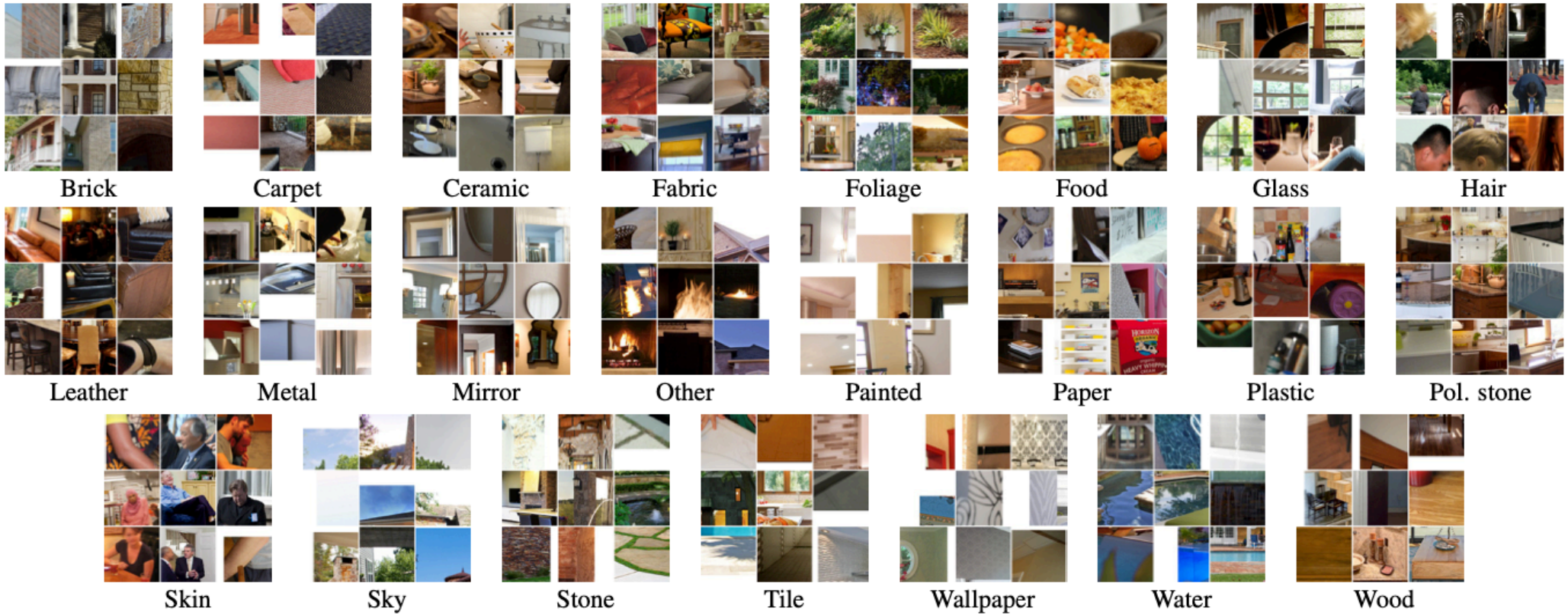


Figure from “A Wholistic View of Deep Neural Networks”, Mundt et al, Neural Networks, 2023

Is including prior knowledge an alternative?



Bell et al, “Material Recognition in the Wild with the Materials in Context Database”, CVPR 2015

Is including prior knowledge an alternative?

Bell et al, “Material Recognition in the Wild with the Materials in Context Database”, CVPR 2015

Leveraging prior knowledge: in essence

Include “**background**” / “**non-examples**” that aren’t of interest

Key questions and considerations:
1. Whether/how to implement the loss?
2. Whether/how to modify embeddings?
3. Whether/how to calibrate post-training?

- What part of the “universe” is useful?
  (“Inference with the universum”, Weston et al, ICML 2006)
- What are we expecting to see during prediction later?
  (Noise, corruption, other concepts?)

1. Modifying the loss with “non-examples”

Example 1: We could let our predictions (classifier) explicitly follow a uniform distribution for “out” data

(Kimin Lee et al, “Training confidence-calibrated classifiers for detecting out-of-distribution samples”, ICLR 2018)

$$\min_{\theta} \mathbb{E}_{P_{\text{in}}(\hat{\mathbf{x}},\hat{y})}\left[-\log P_\theta\left(y=\hat{y}|\hat{\mathbf{x}}\right)\right] + \beta\mathbb{E}_{P_{\text{out}}(\mathbf{x})}\left[KL\left(\mathcal{U}\left(y\right) \parallel P_\theta\left(y|\mathbf{x}\right)\right)\right]$$

1. Modifying the loss with “non-examples”

Example 1: We could let our predictions (classifier) explicitly follow a uniform distribution for “out” data

(Kimin Lee et al, “Training confidence-calibrated classifiers for detecting out-of-distribution samples”, ICLR 2018)

$$\min_{\theta} \mathbb{E}_{P_{\text{in}}(\hat{\mathbf{x}},\hat{y})}\left[-\log P_\theta\left(y=\hat{y}|\hat{\mathbf{x}}\right)\right]+\beta\mathbb{E}_{P_{\text{out}}(\mathbf{x})}\left[KL\left(\mathcal{U}\left(y\right)\parallel P_\theta\left(y|\mathbf{x}\right)\right)\right]$$

Example 2: and many other versions to modify our loss and achieve similar outcome (squish probabilities, make them uniform etc.)

(e.g. Dhamija et al, “Reducing network agnostophobia”, NeurIPS 2018)

$$J_E(x)=\begin{cases}-\log S_c(x) & \text{if } x\in\mathcal{D}'_c \text{ is from class } c\\ -\frac{1}{C}\sum_{c=1}^{C}\log S_c(x) & \text{if } x\in\mathcal{D}'_b\end{cases}$$

2. Calibrating confidence post-hoc

We could calibrate outputs, e.g. by scaling a temperature parameter

(Liang et al, “Enhancing the reliability of out-of-distribution image detection in neural networks”, ICLR 2018)

$$S_i(\boldsymbol{x}; T) = \frac{\exp\left(f_i(\boldsymbol{x})/T\right)}{\sum_{j=1}^{N} \exp\left(f_j(\boldsymbol{x})/T\right)}$$

What does smaller/larger T do in practice?

2. Calibrating confidence post-hoc

We could calibrate outputs, e.g. by scaling a temperature parameter

(Liang et al, “Enhancing the reliability of out-of-distribution image detection in neural networks”, ICLR 2018)

$$S_i(\boldsymbol{x}; T) = \frac{\exp\left(f_i(\boldsymbol{x})/T\right)}{\sum_{j=1}^{N} \exp\left(f_j(\boldsymbol{x})/T\right)}$$

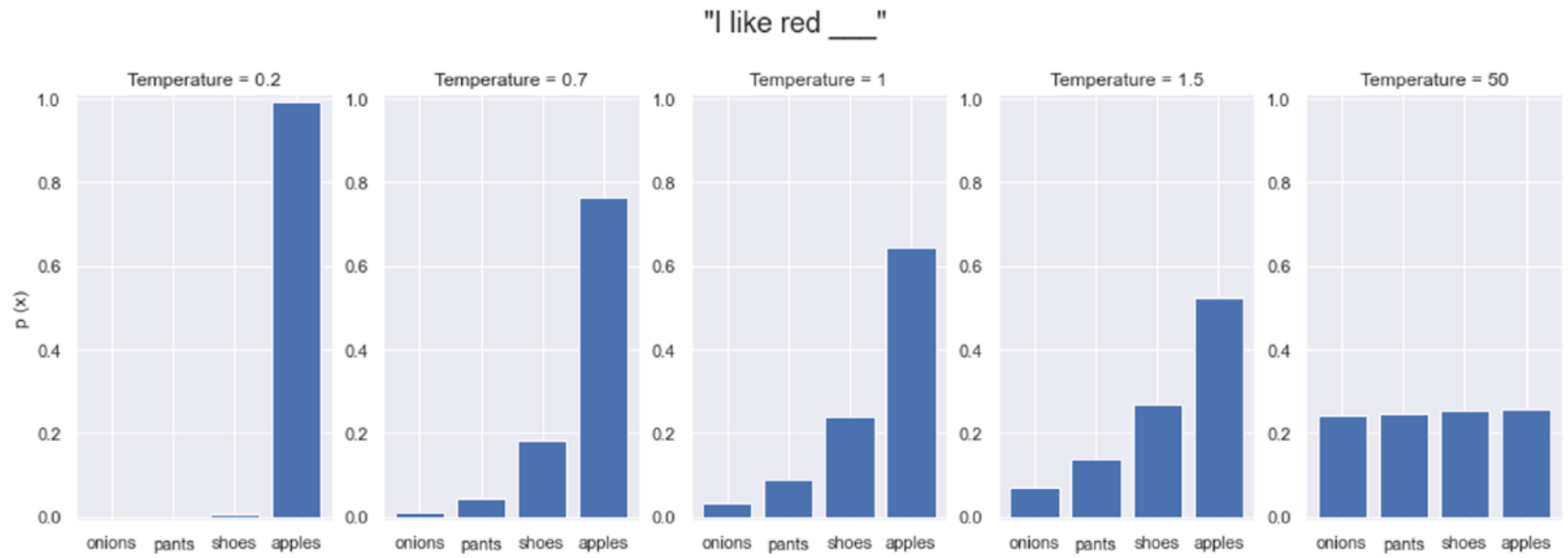


Figure from https://www.hopsworks.ai/dictionary/llm-temperature

2. Calibrating confidence post-hoc

We could calibrate outputs, e.g. by scaling a temperature parameter

(Liang et al, “Enhancing the reliability of out-of-distribution image detection in neural networks”, ICLR 2018)

$$S_i(\boldsymbol{x}; T) = \frac{\exp\left(f_i(\boldsymbol{x})/T\right)}{\sum_{j=1}^{N} \exp\left(f_j(\boldsymbol{x})/T\right)}$$

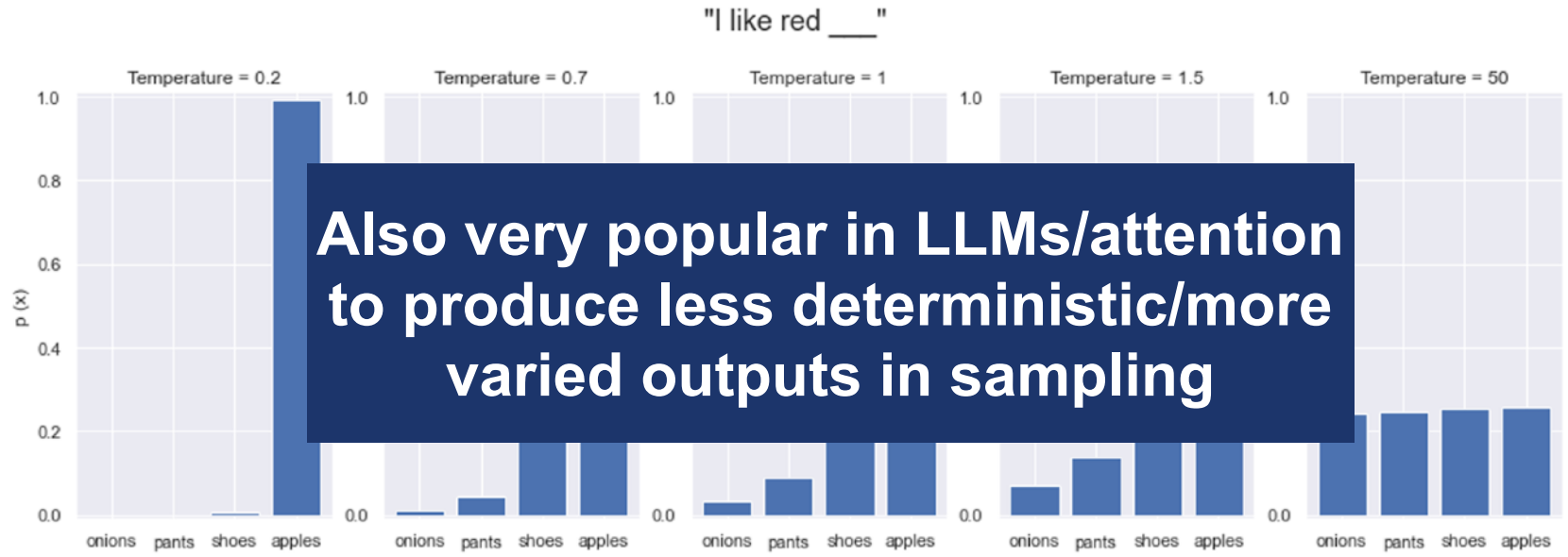
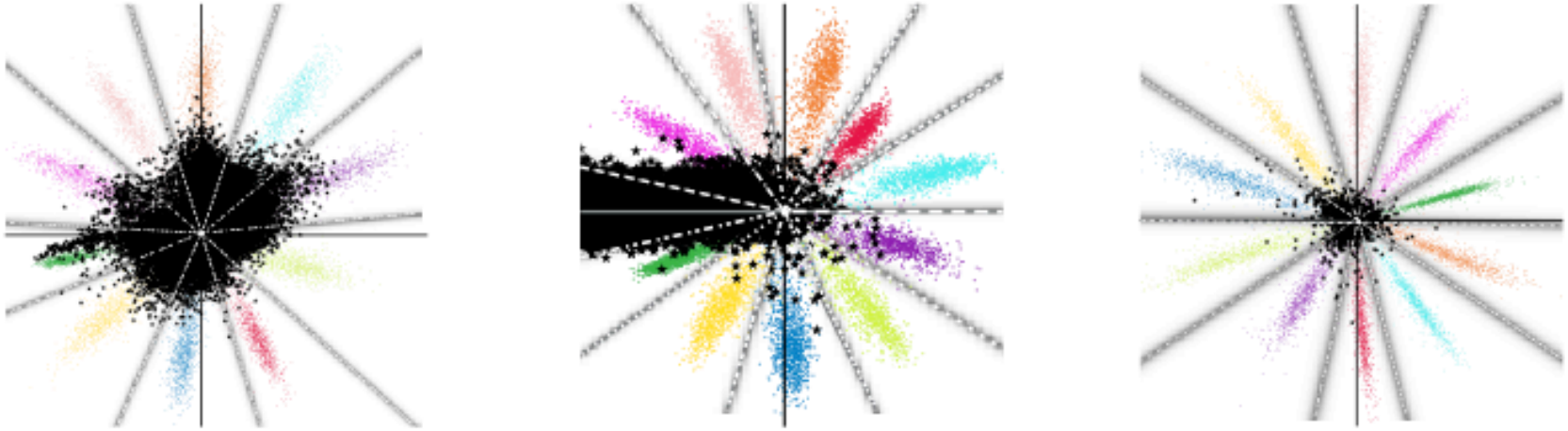


Figure from https://www.hopsworks.ai/dictionary/llm-temperature

3. Steering embeddings with “non-examples”

We could steer embeddings for in- vs. out-of-distribution data -
in the example encouraging features to be zero for OOD data



(a) Softmax (b) Background (c) Objectosphere

Figure 1: LeNet++ Responses To Knowns and Unknowns. *The network in (a) was only trained to classify the 10 MNIST classes ($\mathcal{D}'_c$) using softmax, while the networks in (b) and (c) added NIST letters [15] as known unknowns ($\mathcal{D}'_b$) trained with softmax or our novel Objectosphere loss.*

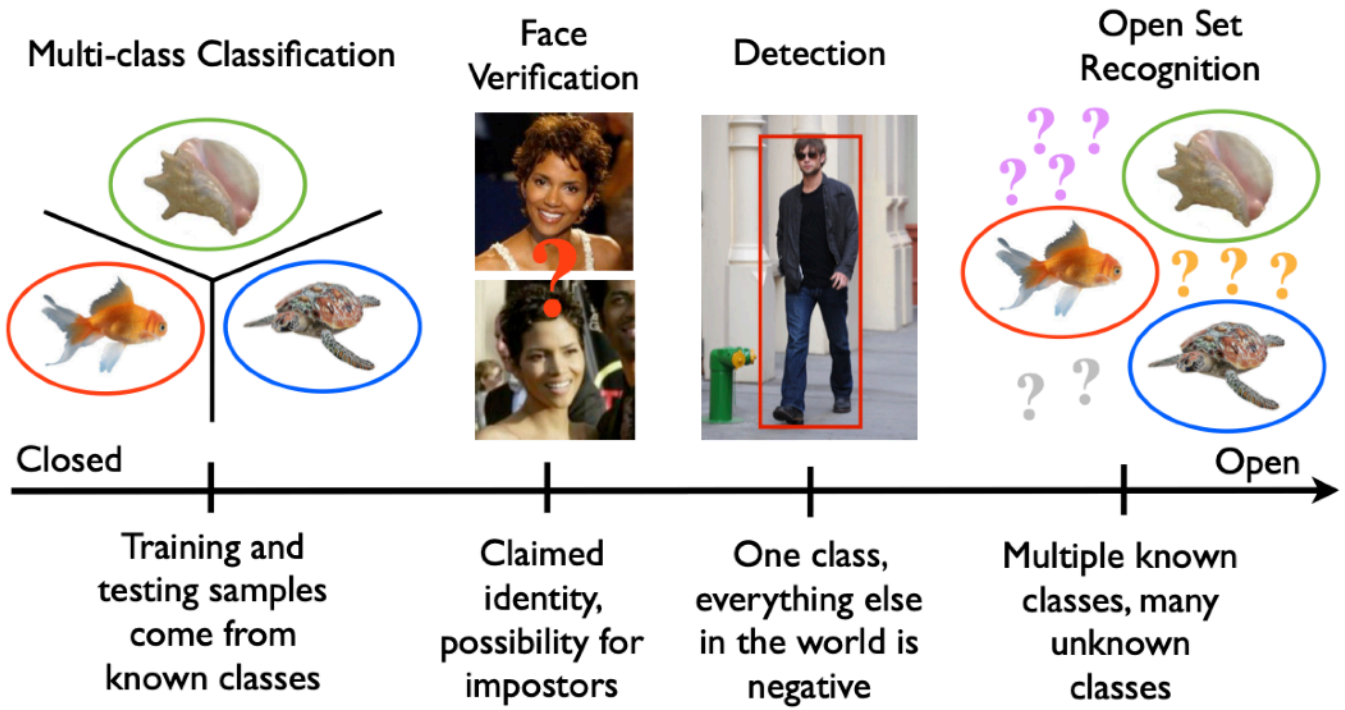
Figure from: Dhamija et al, “Reducing Network Agnostophobia”, NeurIPS 2018

Question time

What do you think are the upsides and downsides so far? Which of our 4 known/unknown cases are we able to address so far?

Closed and open world assumptions

So far, we can only handle knowns and known unknowns reliably!



We need an additional perspective to address unknown unknowns and handle a fully "open" world

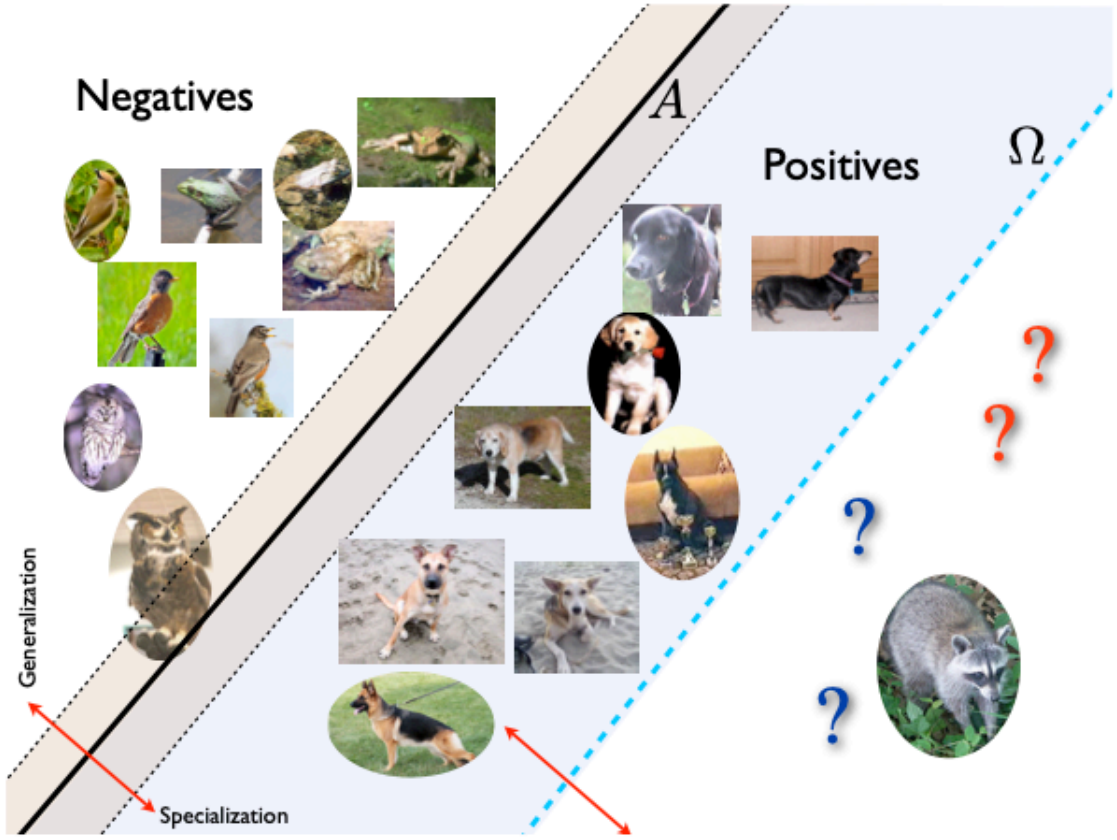
Figure from Scheirer et al, "Towards Open Set Recognition", TPAMI 2012

Examples to build intuition for open space

Example 1: SVM

We could take a second "support set" based on the most distant observed data points

At the hand of these points we fit a second "reject" hyperplane that is parallel to our regular one

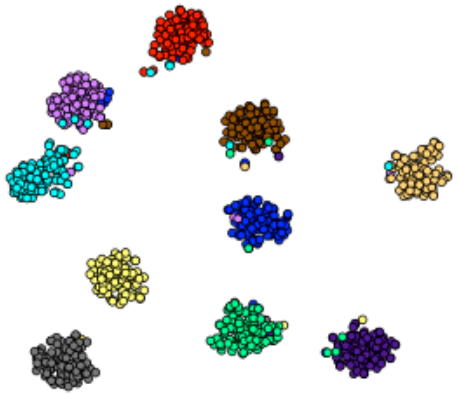


Scheirer et al, "Towards Open Set Recognition", TPAMI 2012

Examples to build intuition for open space

Example 2: normal distributed clusters + distance measures

We could make the assumption/ enforce our data to follow Gaussian clustered embeddings

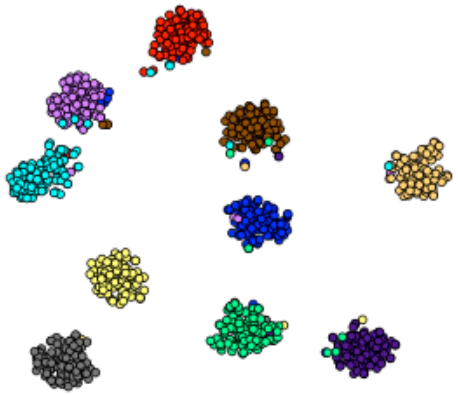


Scheirer et al, “Towards Open Set Recognition”, TPAMI 2012

Examples to build intuition for open space

Example 2: normal distributed clusters + distance measures

We could make the assumption/ enforce our data to follow Gaussian clustered embeddings
-> distance/variance as a test



$$\widehat{\mu}_c = \frac{1}{N_c} \sum_{i:y_i=c} f(\mathbf{x}_i), \quad \widehat{\mathbf{\Sigma}} = \frac{1}{N} \sum_{c} \sum_{i:y_i=c} (f(\mathbf{x}_i) - \widehat{\mu}_c)(f(\mathbf{x}_i) - \widehat{\mu}_c)^\top$$

$$M(\mathbf{x}) = \max_c \; -(f(\mathbf{x}) - \widehat{\mu}_c)^\top \widehat{\mathbf{\Sigma}}^{-1} (f(\mathbf{x}) - \widehat{\mu}_c)$$

Scheirer et al, “Towards Open Set Recognition”, TPAMI 2012

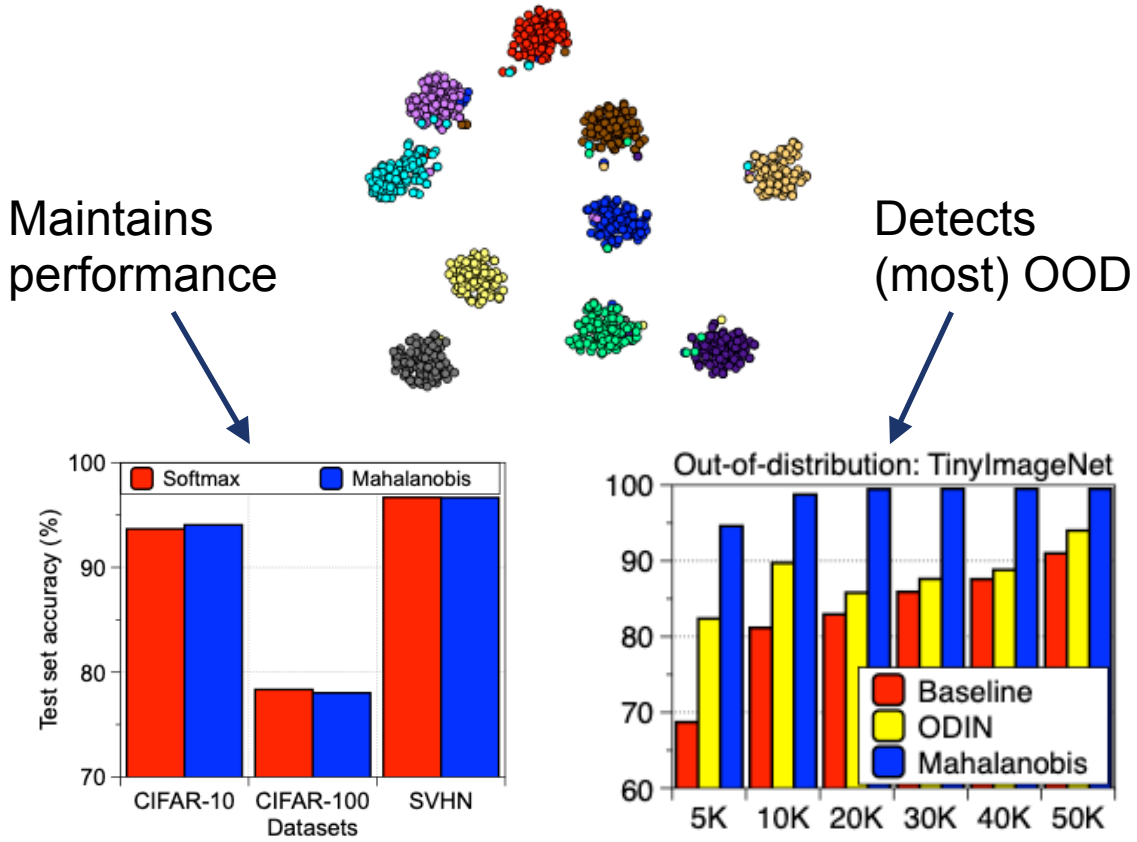
Examples to build intuition for open space

Example 2: normal distributed clusters + distance measures

We could make the assumption/ enforce our data to follow Gaussian clustered embeddings -> distance/variance as a test

$$\widehat{\mu}_c = \frac{1}{N_c} \sum_{i:y_i=c} f(\mathbf{x}_i), \quad \widehat{\mathbf{\Sigma}} = \frac{1}{N} \sum_{c} \sum_{i:y_i=c} (f(\mathbf{x}_i) - \widehat{\mu}_c)(f(\mathbf{x}_i) - \widehat{\mu}_c)^\top$$

$$M(\mathbf{x}) = \max_c \; -(f(\mathbf{x}) - \widehat{\mu}_c)^\top \widehat{\mathbf{\Sigma}}^{-1} (f(\mathbf{x}) - \widehat{\mu}_c)$$



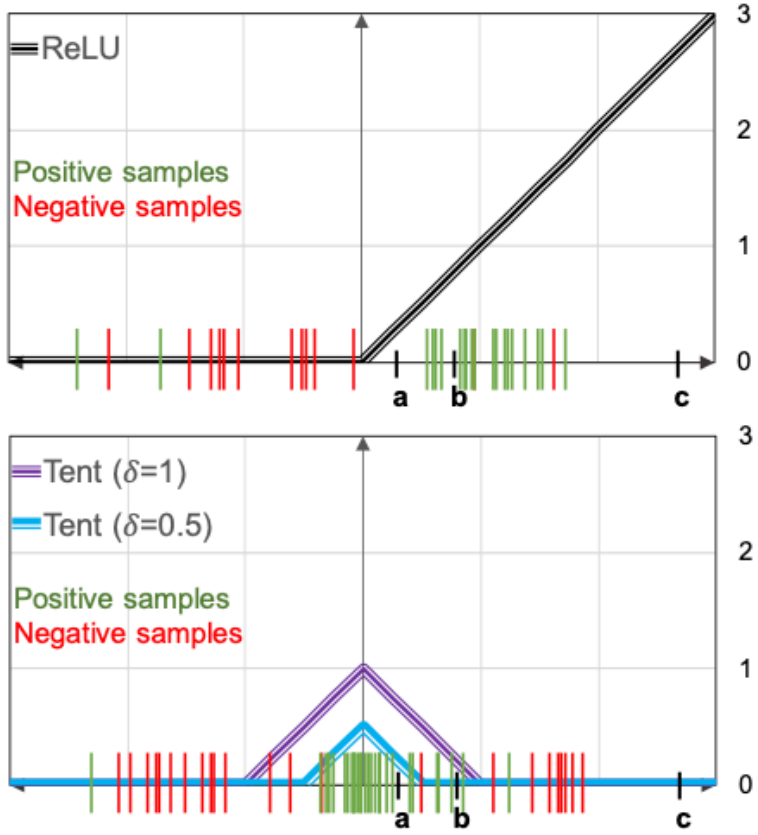
Scheirer et al, “Towards Open Set Recognition”, TPAMI 2012

Examples to build intuition for open space

Example 3: bounded activations

We could define a bounded activation function and learn its "extent" based on observed data, e.g. a "tent" like function

$$f(x;\delta) = \max(0, \delta - |x|)$$



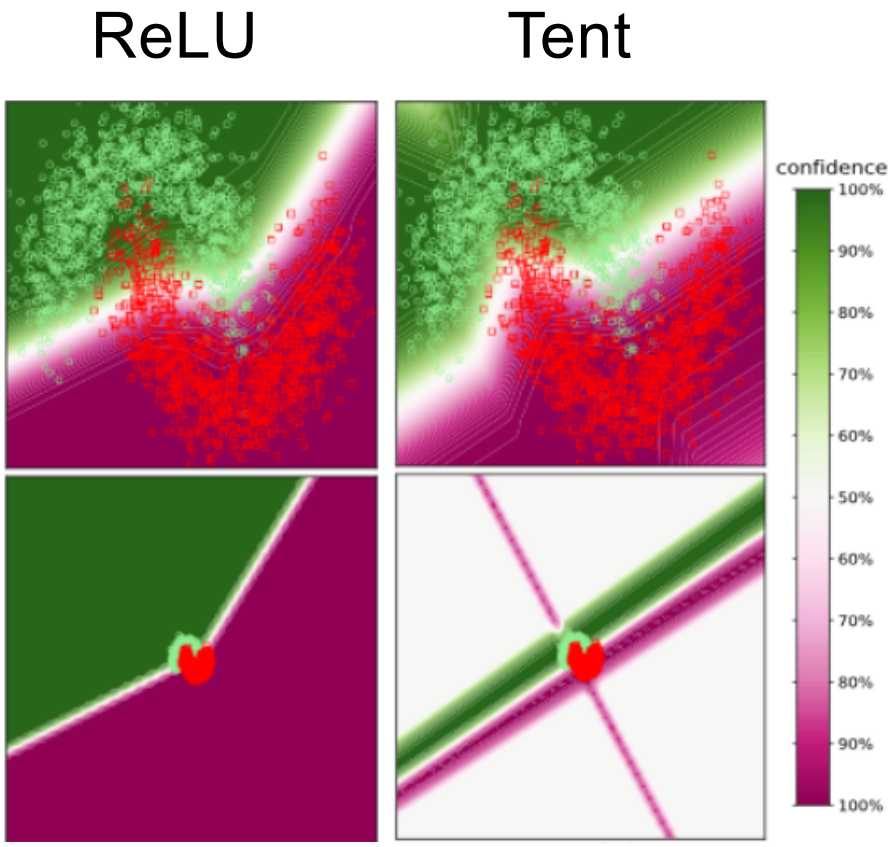
Rozsa & Boult, "Improved Adversarial Robustness by Reducing Open Space Risk via Tent Activations", 2019

Examples to build intuition for open space

Example 3: bounded activations

We could define a bounded activation function and learn its “extent” based on observed data, e.g. a “tent” like function

$$f(x;\delta) = \max(0, \delta - |x|)$$

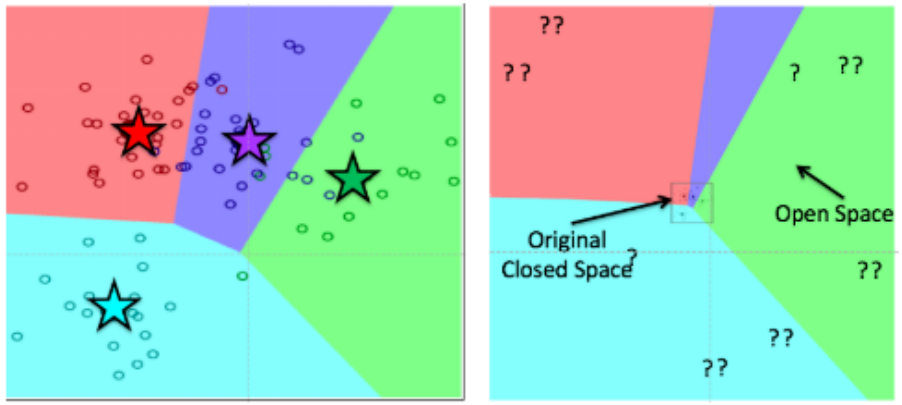


Rozsa & Boult, “Improved Adversarial Robustness by Reducing Open Space Risk via Tent Activations”, 2019, Figure reproduced by Quentin Delfosse

Universität Bremen

Formalizing open space

Intuitively: what is not covered with data

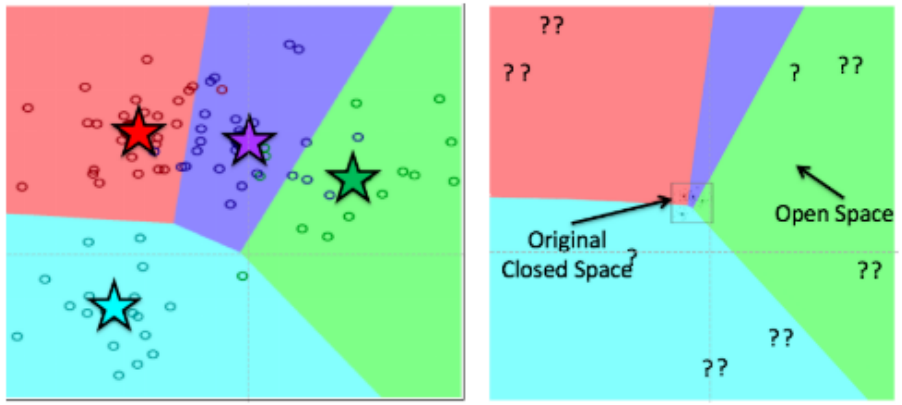


“Learning and the Unknown”, Boult et al, AAAI 2019,

Formalizing open space

Intuitively: what is not covered with data

Formally: For a recognition function f over space $\mathcal{X}$ & a union of balls B with radius r that includes all known training examples: $\mathcal{O} = \mathcal{X} - \cup_{i \in N} B_r(x_i)$

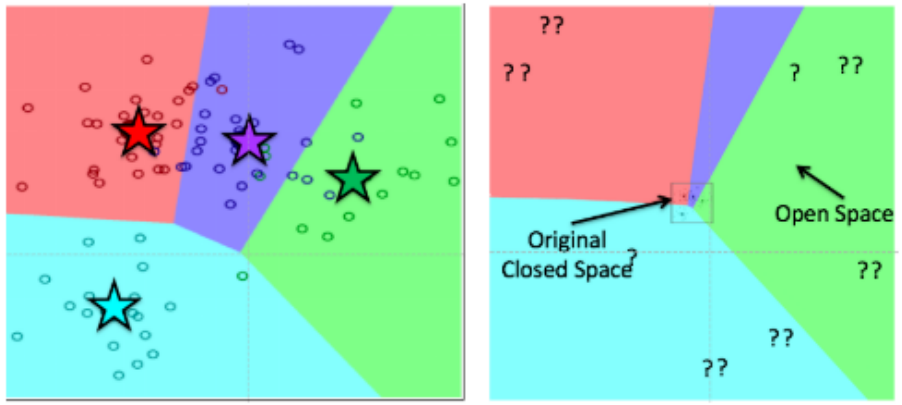


“Learning and the Unknown”, Boult et al, AAAI 2019,

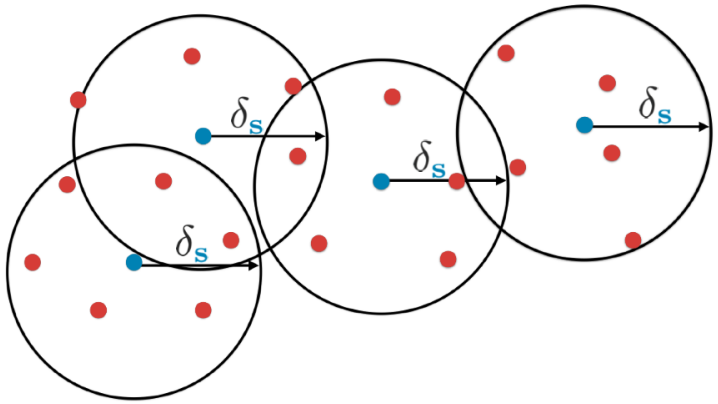
Formalizing open space

Intuitively: what is not covered with data

Formally: For a recognition function f over space $\mathcal{X}$ & a union of balls B with radius r that includes all known training examples: $\mathcal{O} = \mathcal{X} - \cup_{i \in N} B_r(x_i)$



Recall: *"a set s is a δ cover of a set $s^\star$ means a set of balls with radius δ centered at each member of s can cover the entire $s^\star$"*



"Learning and the Unknown", Boult et al, AAAI 2019,
Sener & Savarese, Active Learning for CNNs: A Core-set Approach, ICLR 2018

Formalizing open space risk

We can now define open space risk (OSR) as: $R_{\mathcal{O}}(f) = \frac{\int_{\mathcal{O}} f(x)dx}{\int_{S_V} f(x)dx}$

where S_V is the closed set (set of all points within $\mathcal{X}$ covered by the balls

Formalizing open space risk

We can now define open space risk (OSR) as: $R_{\mathcal{O}}(f) = \frac{\int_{\mathcal{O}} f(x)dx}{\int_{S_V} f(x)dx}$

where S_V is the closed set (set of all points within $\mathcal{X}$ covered by the balls

Example: we can reduce OSR by minimizing the volume of the indicator function on the right

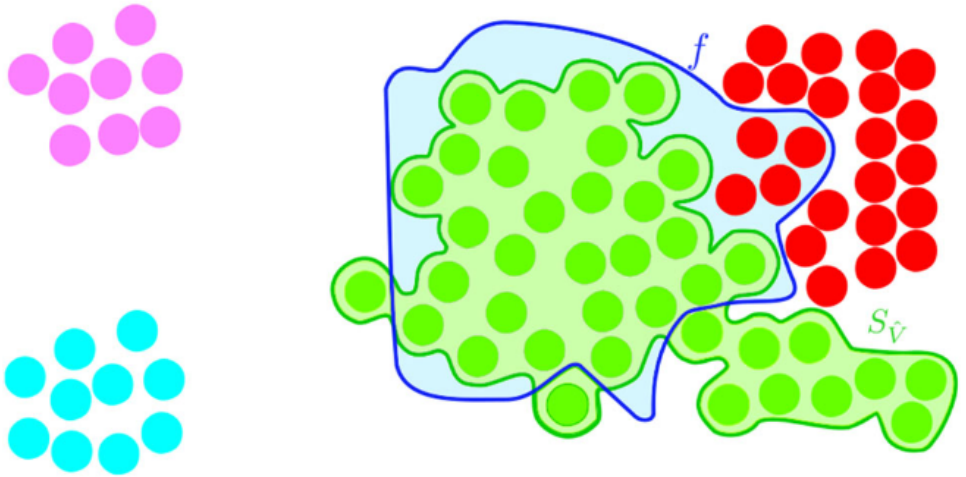
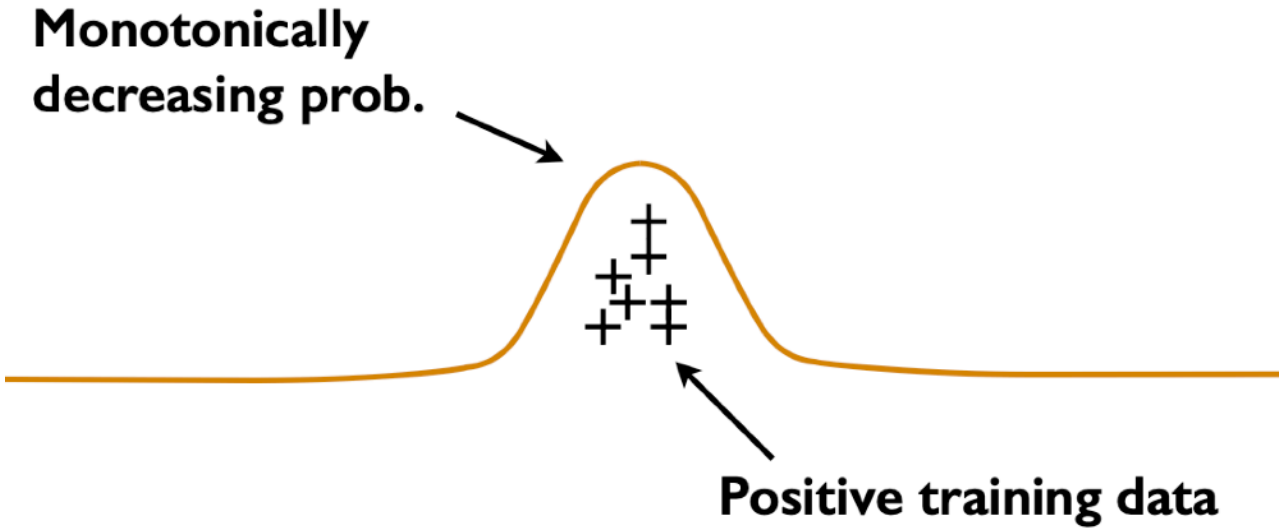


Fig. 2 Conceptual illustration of OSR: the closed set $S_{\hat{V}}$ is visualized by the green area comprising all inliers. There are two different rest samples types: red points resemble a known rest class, whereas blue and pink samples reflect unobserved outliers. The objective is to learn an indicator function f (decision boundary shown as blue line) that filters inliers and rejects all rest samples. The open space risk $R_{\mathcal{O}}(f)$ can be interpreted as the ratio of the blue area over the green area that is bounded by the decision boundary of f (color figure online)

Figure from Lüberring et al, “Bounding open space risk with decoupling auto encoders in open set recognition”, Journal of Data Science and Analytics 14, 2022

Open space and open space risk

As an intuitive summary, we try to approximate the space that is supported by our observed data (and the interpolation region between these points) and construct a measure which decays probability away from our supporting evidence



Scheirer et al, “Probability Models for Open Set Recognition”, TPAMI 2014

OSR in deep neural networks: OpenMax

A first “deep” algorithm:

- Aggregates activations
- Doesn’t directly implement the “ball cover” idea, but instead computes a single mean per category & computes each data points’ distances

Algorithm 1 EVT Meta-Recognition Calibration for Open Set Deep Networks,

Require: FitHigh function from libMR
Require: Activation levels in the penultimate network layer $\mathbf{v}(\mathbf{x}) = v_1(x) \ldots v_N(x)$
Require: For each class j let $S_{i,j} = v_j(x_{i,j})$ for each correctly classified training example $x_{i,j}$.
1: **for** $j = 1 \ldots N$ **do**
2: **Compute mean AV**, $\mu_j = mean_i(S_{i,j})$

Bendale & Boult et al, “Towards Open Set Deep Networks”, CVPR 2016

OSR in deep neural networks: OpenMax

- Fits a statistical model to reject based on this way of measuring observed “coverage”
- Key idea is that we don’t care about the mean all that much for OOD detection, but what the largest known distances are

But what kind of distribution should we fit?

Algorithm 1 EVT Meta-Recognition Calibration for Open Set Deep Networks,

Require: FitHigh function from libMR
Require: Activation levels in the penultimate network layer $\mathbf{v}(\mathbf{x}) = v_1(x) \ldots v_N(x)$
Require: For each class j let $S_{i,j} = v_j(x_{i,j})$ for each correctly classified training example $x_{i,j}$.

1: **for** $j = 1 \ldots N$ **do**
2: **Compute mean AV**, $\mu_j = mean_i(S_{i,j})$
3: **EVT Fit** $\rho_j = (\tau_j, \kappa_j, \lambda_j) = \text{FitHigh}(\|\hat{S}_j - \mu_j\|, \eta)$
4: **end for**
5: **Return** means μ_j and libMR models ρ_j

Bendale & Boult et al, “Towards Open Set Deep Networks”, CVPR 2016

OSR in deep neural networks: OpenMax

- Fits a statistical model to reject based on this way of measuring observed “coverage”
- Key idea is that we don’t care about the mean all that much for OOD detection, but what the largest known distances are

The extreme value theorem provides us with an answer!

Algorithm 1 EVT Meta-Recognition Calibration for Open Set Deep Networks, with per class Weibull fit to η largest distance to mean activation vector. Returns libMR models ρ_j which includes parameters τ_i for shifting the data as well as the Weibull shape and scale parameters:κ_i, λ_i.

Require: FitHigh function from libMR

Require: Activation levels in the penultimate network layer $\mathbf{v}(\mathbf{x}) = v_1(x) \ldots v_N(x)$

Require: For each class j let $S_{i,j} = v_j(x_{i,j})$ for each correctly classified training example $x_{i,j}$.

1: **for** $j = 1 \ldots N$ **do**
2: **Compute mean AV**, $\mu_j = mean_i(S_{i,j})$
3: **EVT Fit** $\rho_j = (\tau_j, \kappa_j, \lambda_j) = \text{FitHigh}(\|\hat{S}_j - \mu_j\|, \eta)$
4: **end for**
5: **Return** means μ_j and libMR models ρ_j

Bendale & Boult et al, “Towards Open Set Deep Networks”, CVPR 2016

Central limit vs. extreme value theorem

2 fundamental concepts in statistics

- The **central limit theorem** focuses on the normality of means
  -> given large enough sample size the distribution of means is approximately normal distributed

Central limit vs. extreme value theorem

2 fundamental concepts in statistics

- The **central limit theorem** focuses on the normality of means
  -> given large enough sample size the distribution of means is approximately normal distributed
- The **extreme value theorem** focuses on minima/maxima that govern the tails of any distribution
  -> the extremes of a sample of iid random variables converges to one of three possible distributions

Central limit vs. extreme value theorem

2 fundamental concepts in statistics

- The **central limit theorem** focuses on the normality of means *-> given large enough sample size the distribution of means is approximately normal distributed*
- The **extreme value theorem** focuses on minima/maxima that govern the tails of any distribution *-> the extremes of a sample of iid random variables converges to one of three possible distributions*

GEV distribution type I,II,III: Gumbel, Fréchet, Weibull (all some form of exponential)

$$f(x;\lambda,k) = \begin{cases} \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

Central limit vs. extreme value theorem

2 fundamental concepts in statistics

- The **central limit theorem** focuses on the normality of means *-> given large enough sample size the distribution of means is approximately normal distributed*
- The **extreme value theorem** focuses on minima/maxima that govern the tails of any distribution *-> the extremes of a sample of iid random variables converges to one of three possible distributions*

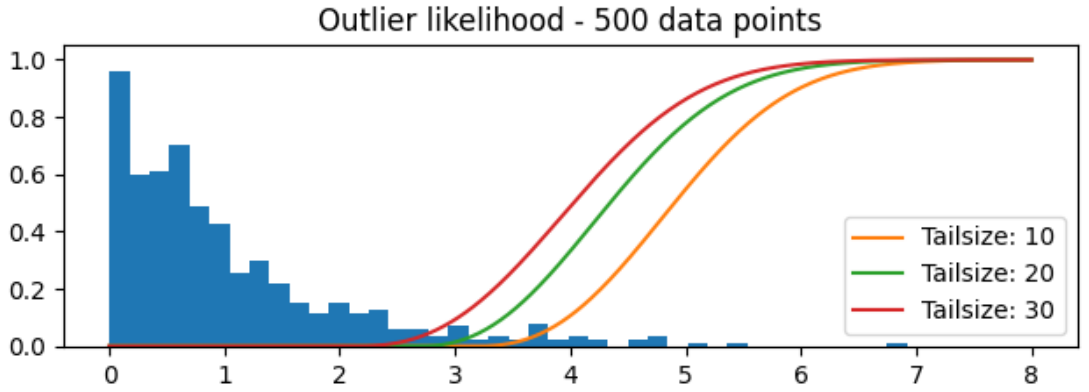
GEV distribution type I,II,III: Gumbel, Fréchet, Weibull (all some form of exponential)

$$f(x;\lambda,k) = \begin{cases} \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

Extreme value theory is interested in the probability of events that are more extreme than any previously observed

Central limit vs. extreme value theorem

We can make use of the cumulative distribution function (CDF) to reject data points, if they exceed our expected extremely observed distances



OSR in deep neural networks: OpenMax

We can make use of the cumulative distribution function (CDF) to reject data points, if they exceed our expected extremely observed distances

We can also use this obtained value to modify our output, e.g. “squish” the probability, temperature scale the SoftMax etc. Hence “OpenMax” in our example

Algorithm 2 OpenMax probability estimation with rejection of unknown or uncertain inputs.

Require: Activation vector for $\mathbf{v}(\mathbf{x}) = v_1(x), \ldots, v_N(x)$
Require: **means** μ_j and libMR models $\rho_j = (\tau_i, \lambda_i, \kappa_i)$
Require: α, the numer of “top” classes to revise

1: Let $s(i) = \text{argsort}(v_j(x))$; Let $\omega_j = 1$
2: **for** $i = 1, \ldots, \alpha$ **do**
3: $\omega_{s(i)}(x) = 1 - \frac{\alpha - i}{\alpha} e^{-\left(\frac{\|x - \tau_{s(i)}\|}{\lambda_{s(i)}}\right)^{\kappa_{s(i)}}}$
4: **end for**
5: Revise activation vector $\hat{v}(x) = \mathbf{v}(\mathbf{x}) \circ \omega(\mathbf{x})$
6: Define $\hat{v}_0(x) = \sum_i v_i(x)(1 - \omega_i(x))$.
7:

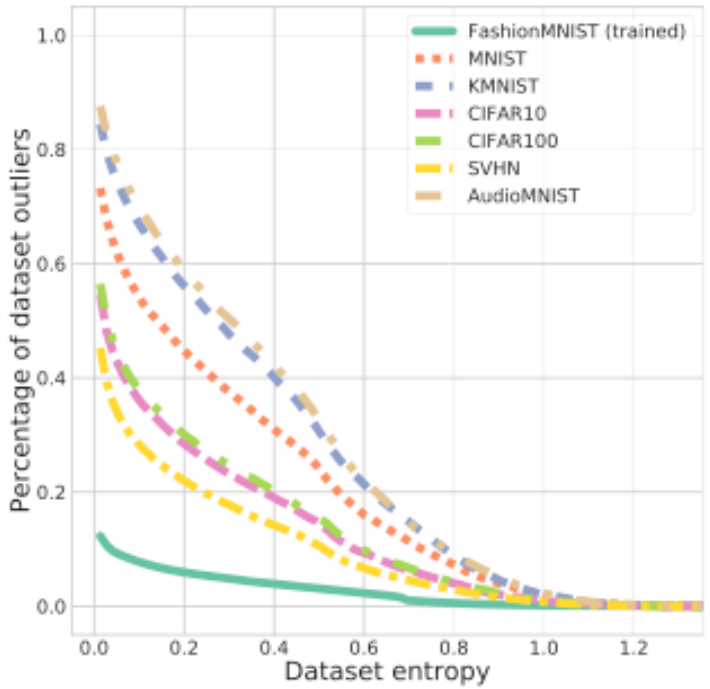
$$\hat{P}(y = j|\mathbf{x}) = \frac{e^{\hat{\mathbf{v}}_\mathbf{j}(\mathbf{x})}}{\sum_{i=0}^{N} e^{\hat{\mathbf{v}}_\mathbf{i}(\mathbf{x})}}$$

8: Let $y^* = \text{argmax}_j \, P(y = j|\mathbf{x})$
9: Reject input if $y^* == 0$ or $P(y = y^*|\mathbf{x}) < \epsilon$

Bendale & Boult et al, “Towards Open Set Deep Networks”, CVPR 2016

OSR in deep neural networks: OpenMax

How much does OpenMax (EVT bounds) improve pure confidence?

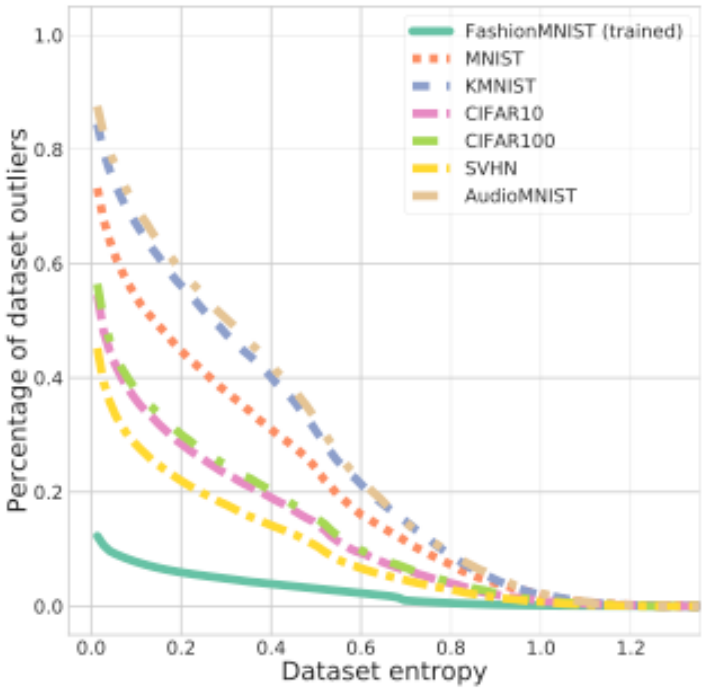


Predictive Entropy

Figure: Mundt et al “Open Set Recognition Through Deep Neural Network Uncertainty”, ICCVW 2019

OSR in deep neural networks: OpenMax

How much does OpenMax (EVT bounds) improve pure confidence?



Predictive Entropy

OpenMax

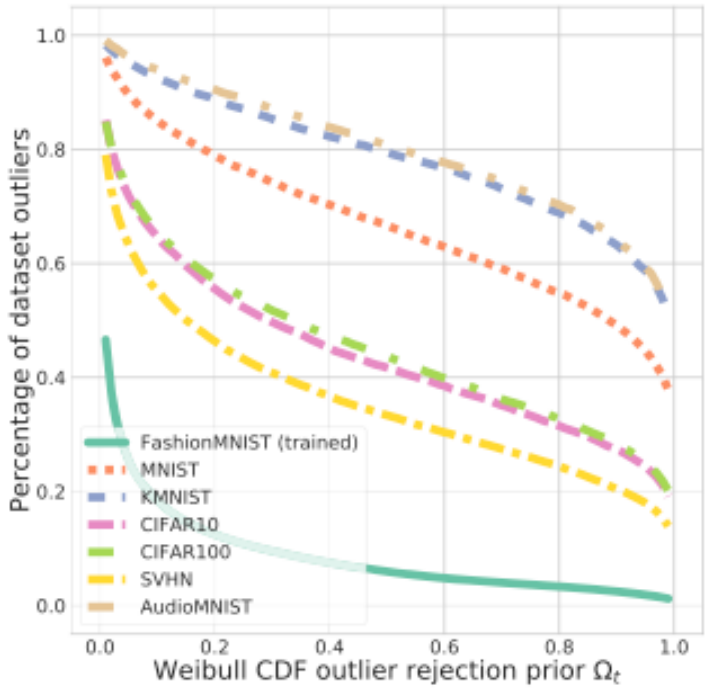
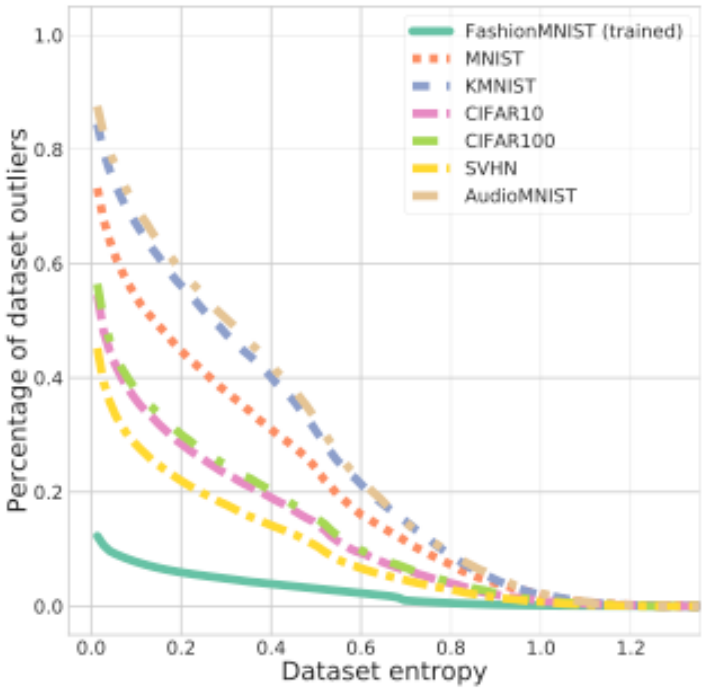


Figure: Mundt et al “Open Set Recognition Through Deep Neural Network Uncertainty”, ICCVW 2019

OSR in deep neural networks: OpenMax

How much does OpenMax (EVT bounds) improve pure confidence?



Predictive Entropy

Why don't we observe even greater improvement?

OpenMax

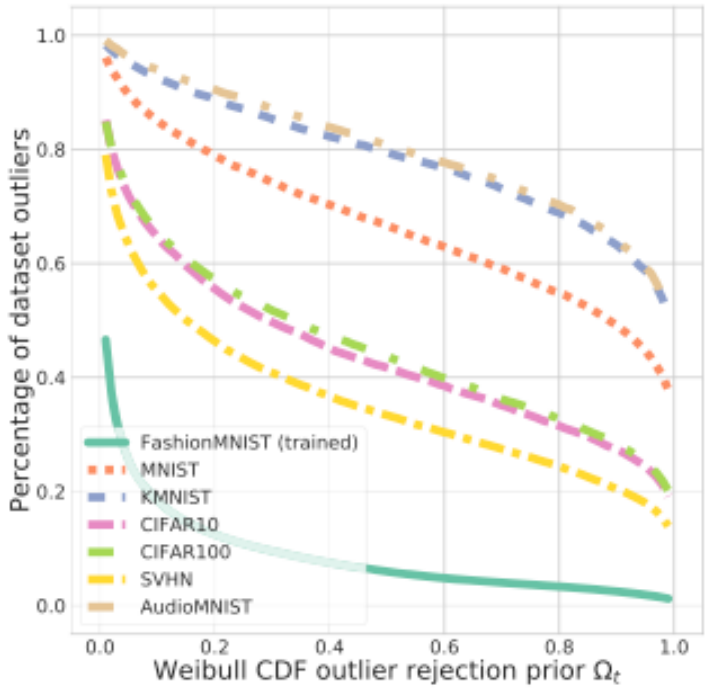
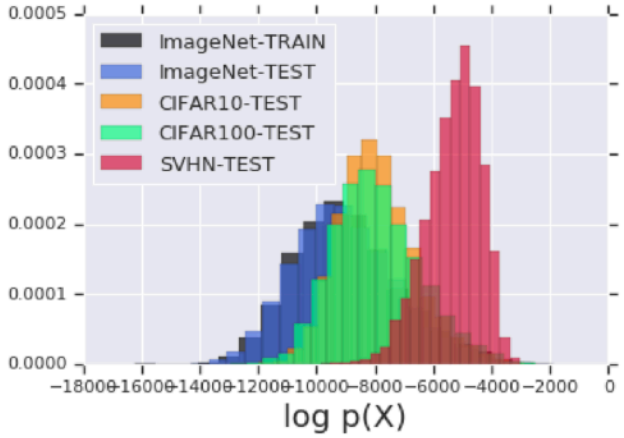


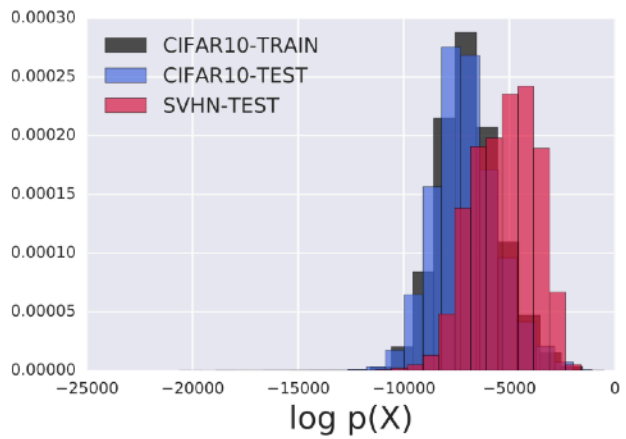
Figure: Mundt et al "Open Set Recognition Through Deep Neural Network Uncertainty", ICCVW 2019

Recall: Generative model predictions

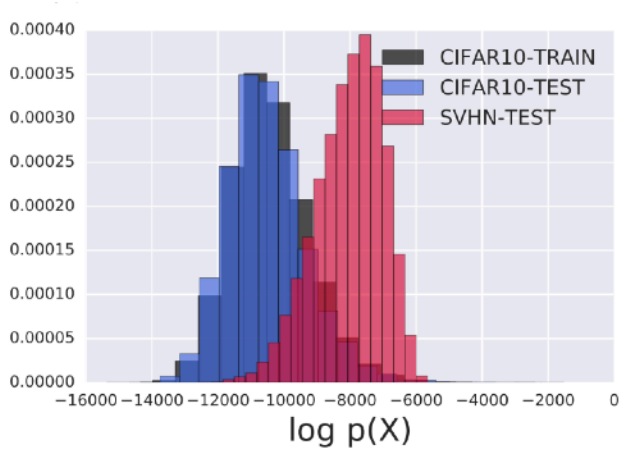
Predictions may be overconfident! But for out-of-distribution detection, we may still need generative models to learn about p(x)!



Glow
(Normalizing Flow)



PixelCNN
(Autoregression)



VAE
(Variational Inference)

Nalisnick et al, “Do Deep Generative Models Know What They Don’t Know”, ICLR 2019

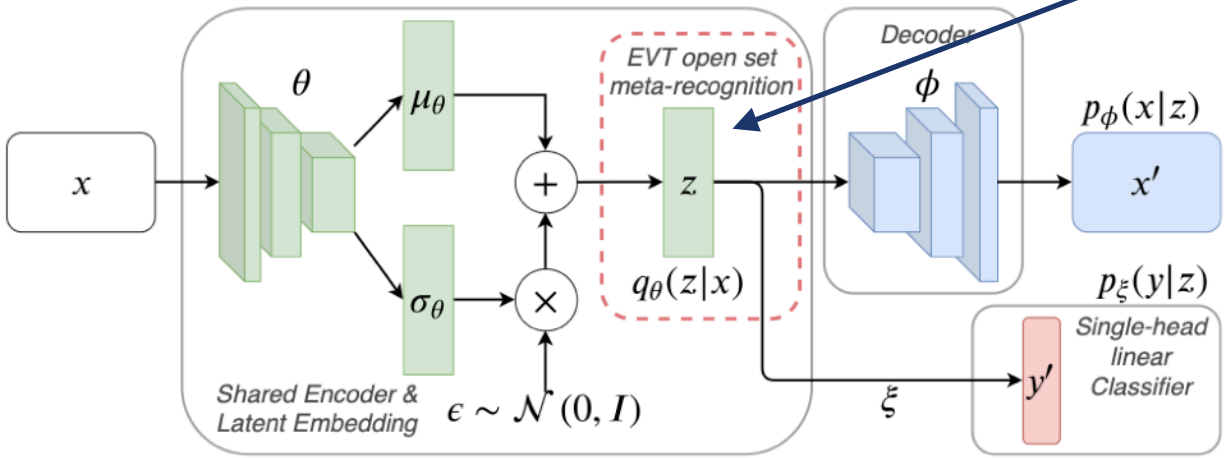
EVT in discriminative & generative models

- Recall: discriminative models may learn may decision shortcuts
- If we don’t explicitly know about p(x), how can we know what isn’t?

EVT in discriminative & generative models

- Recall: discriminative models may learn may decision shortcuts
- If we don't explicitly know about p(x), how can we know what isn't?

Apply what we learned on generative factors z, rather than outputs or arbitrary activations

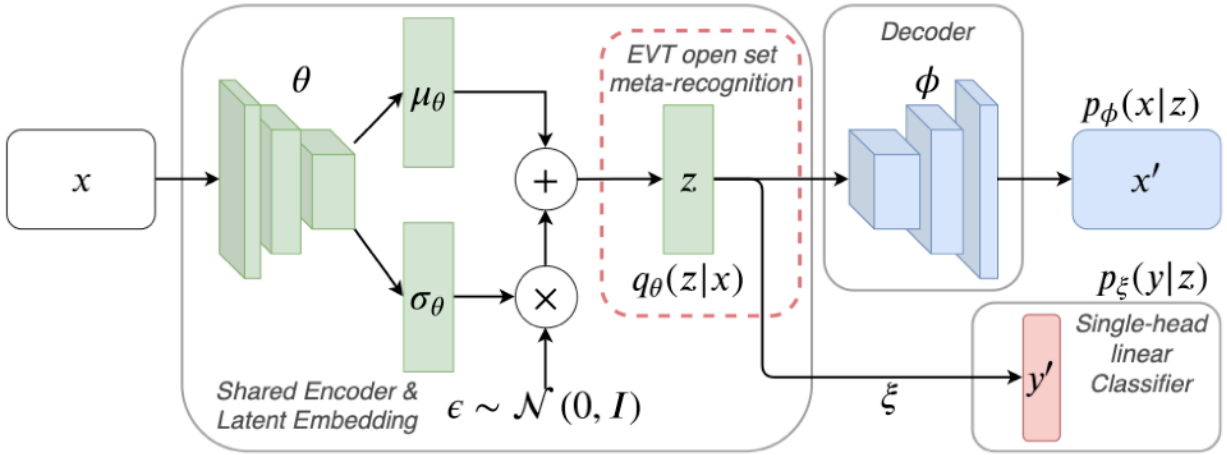


Example: from OpenMax to OpenVAE

Mundt et al, "Unified Probabilistic Deep Continual Learning Through Open Set Recognition and Generative Replay", Journal of Imaging, Volume 8, Issue 4, 2022

EVT in discriminative & generative models

- Recall: discriminative models may learn may decision shortcuts
- If we don't explicitly know about p(x), how can we know what isn't?



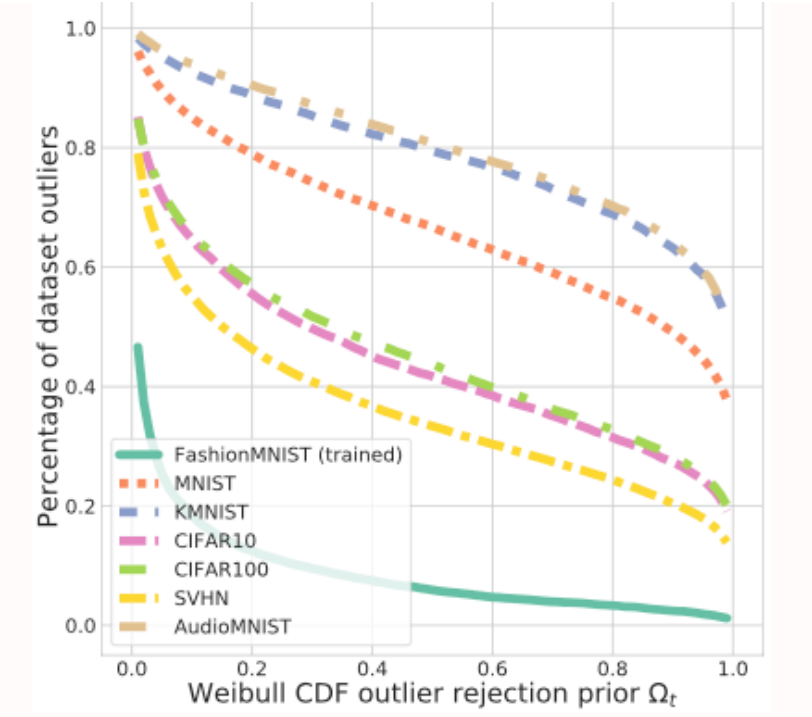
Algorithm 1 Open set recognition calibration for deep variational neural networks. A Weibull model fit of tail-size η is conducted to bound the per class approximate posterior. Per class c Weibull models $\boldsymbol{\rho}_c$ with their respective shift τ_c, shape κ_c and scale λ_c parameters are returned.

Require: Trained encoder $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ and classifier $p_{\boldsymbol{\xi}}(\boldsymbol{y}|\boldsymbol{z})$
Require: Classifier probabilities $p_{\boldsymbol{\xi}}(\boldsymbol{y}|\boldsymbol{z})$ and samples from the approximate posterior $\boldsymbol{z}(\boldsymbol{x}^{(i)}) \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})$ for each training dataset example $\boldsymbol{x}^{(i)}$
Require: For each class c, let $\boldsymbol{S}_c^{(i)} = \boldsymbol{z}(\boldsymbol{x}_c'^{(i)})$ for each correctly classified training example $\boldsymbol{x}_c'^{(i)}$
1: **for** $c = 1 \ldots C$ **do**
2: **Get per class latent mean** $\bar{\boldsymbol{S}}_c = mean(\boldsymbol{S}_c^{(i)})$
3: **Weibull model** $\boldsymbol{\rho}_c =$ Fit Weibull $\left(||\boldsymbol{S}_c - \bar{\boldsymbol{S}}_c||, \eta\right)$
4: **Return** means $\bar{\boldsymbol{S}}$ and Weibull models $\boldsymbol{\rho}$

Mundt et al, "Unified Probabilistic Deep Continual Learning Through Open Set Recognition and Generative Replay", Journal of Imaging, Volume 8, Issue 4, 2022

OSR in deep gen. neural networks: OpenVAE

How much does learning about p(x) help us create useful bounds?

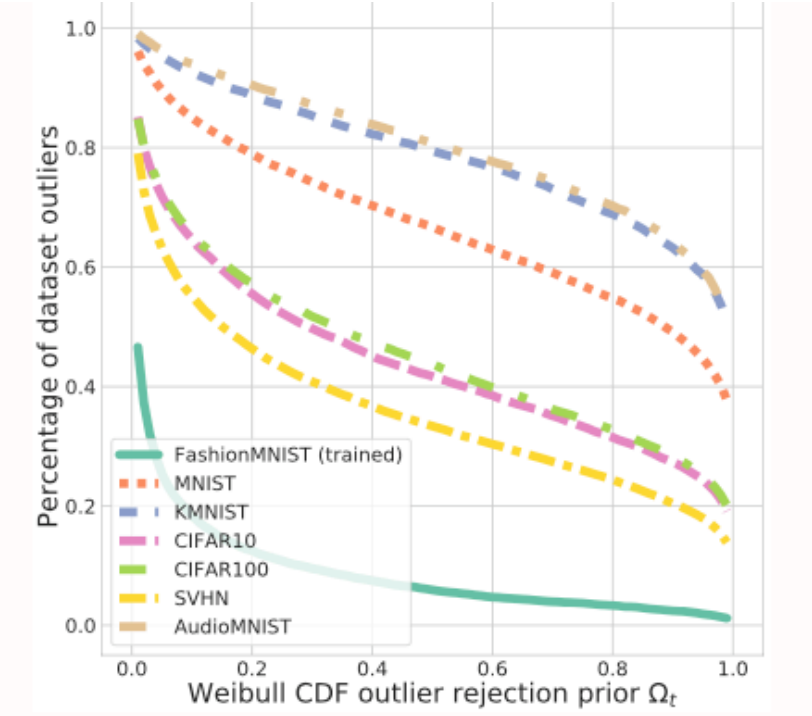


OpenMax

Figure: Mundt et al “Open Set Recognition Through Deep Neural Network Uncertainty”, ICCVW 2019

OSR in deep gen. neural networks: OpenVAE

How much does learning about p(x) help us create useful bounds?



OpenMax

OpenVAE

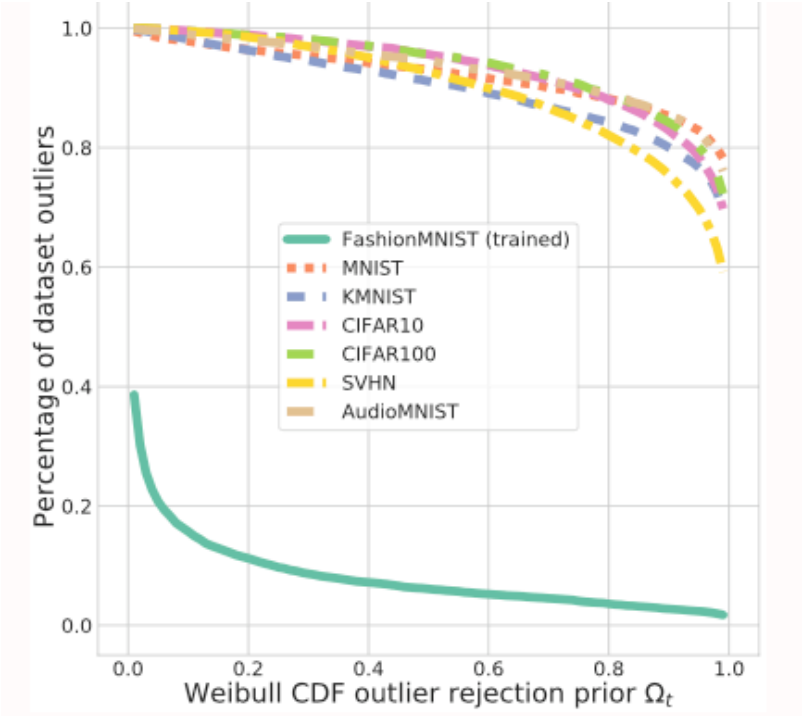
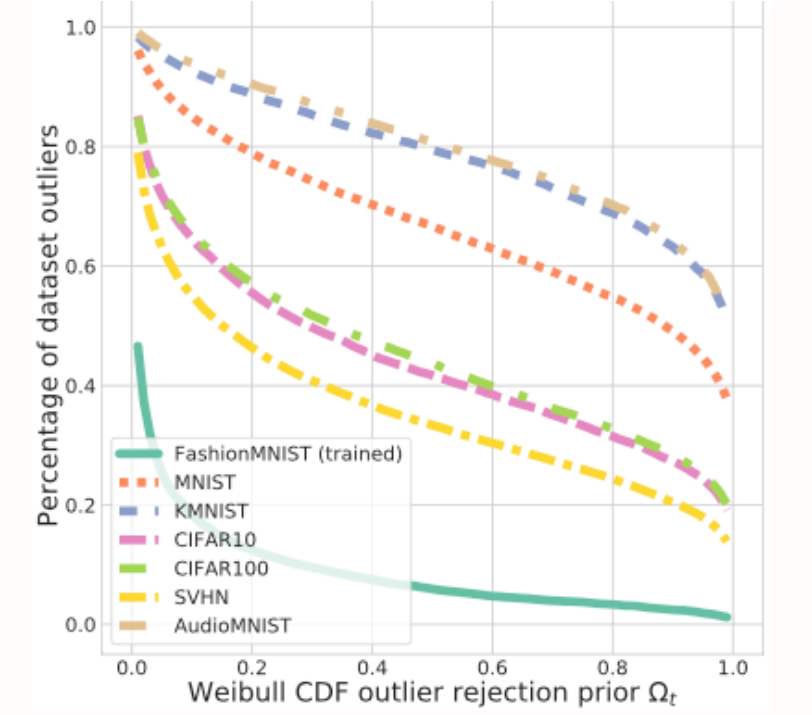


Figure: Mundt et al “Open Set Recognition Through Deep Neural Network Uncertainty”, ICCVW 2019

OSR in deep gen. neural networks: OpenVAE

How much does learning about p(x) help us create useful bounds?



OpenMax

OOD In essence: we need bounds & representations!

OpenVAE

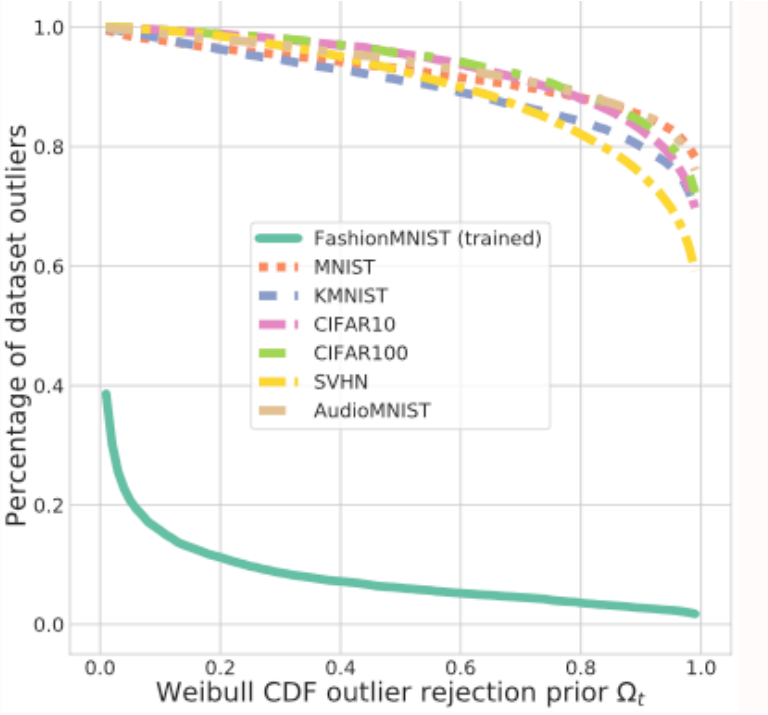


Figure: Mundt et al “Open Set Recognition Through Deep Neural Network Uncertainty”, ICCVW 2019

In retrospect: increments in a closed world

Novel deep architectures seem to improve, but also fall “flat” when benchmarked in the presence of an “open world”

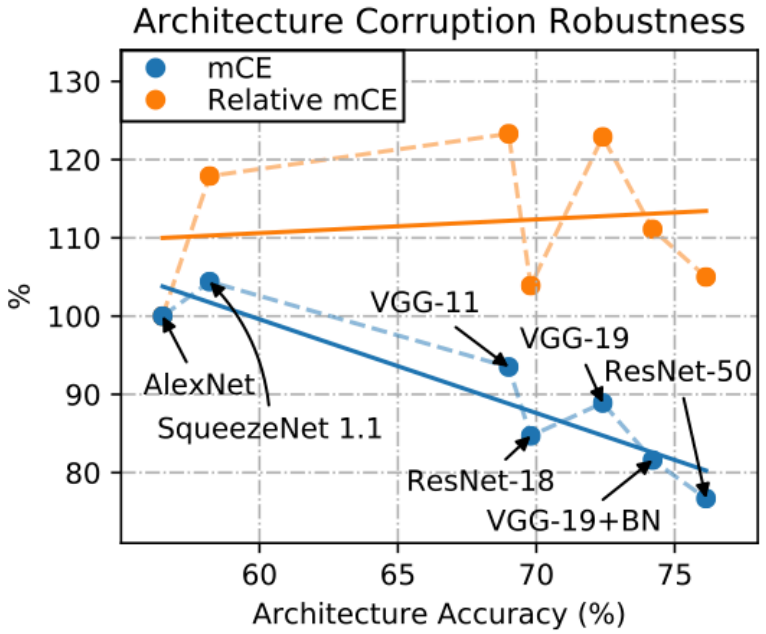


Figure 3: Robustness (mCE) and Relative mCE IMAGENET-C values. Relative mCE values suggest robustness in itself declined from AlexNet to ResNet. “BN” abbreviates Batch Normalization.

Figure from: Hendricks & Dietterich, “Benchmarking Neural Network Robustness to Common Corruptions and Perturbations”, ICLR 2019

In retrospect: increments in a closed world

Novel deep architectures seem to improve, but also fall “flat” when benchmarked in the presence of an “open world”

- “Active learning” doesn’t consider continued training, nor does it consider various data from an open world
- “Curriculum learning” is often studied in isolation from active learning on purely stationary benchmark datasets

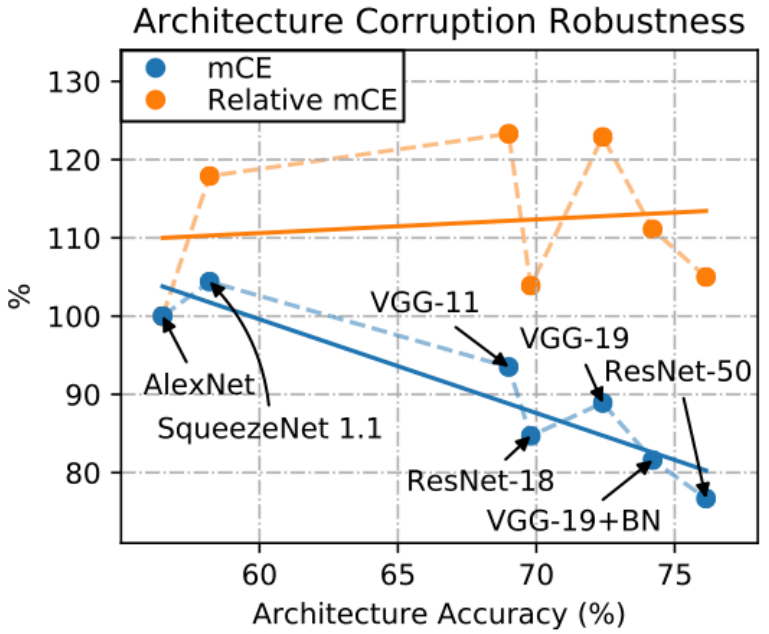


Figure 3: Robustness (mCE) and Relative mCE IMAGENET-C values. Relative mCE values suggest robustness in itself declined from AlexNet to ResNet. “BN” abbreviates Batch Normalization.

Figure from: Hendricks & Dietterich, “Benchmarking Neural Network Robustness to Common Corruptions and Perturbations”, ICLR 2019

In retrospect: increments in a closed world

The closed world assumption in ML is VERY prevalent throughout!

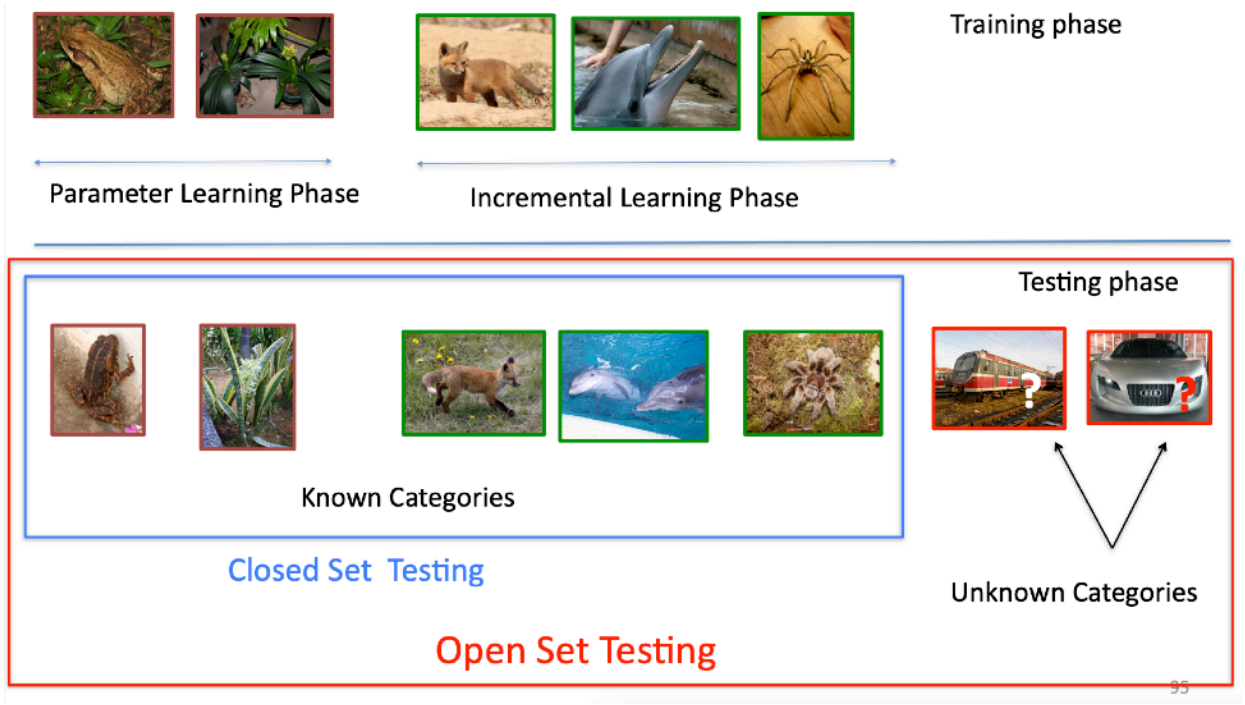
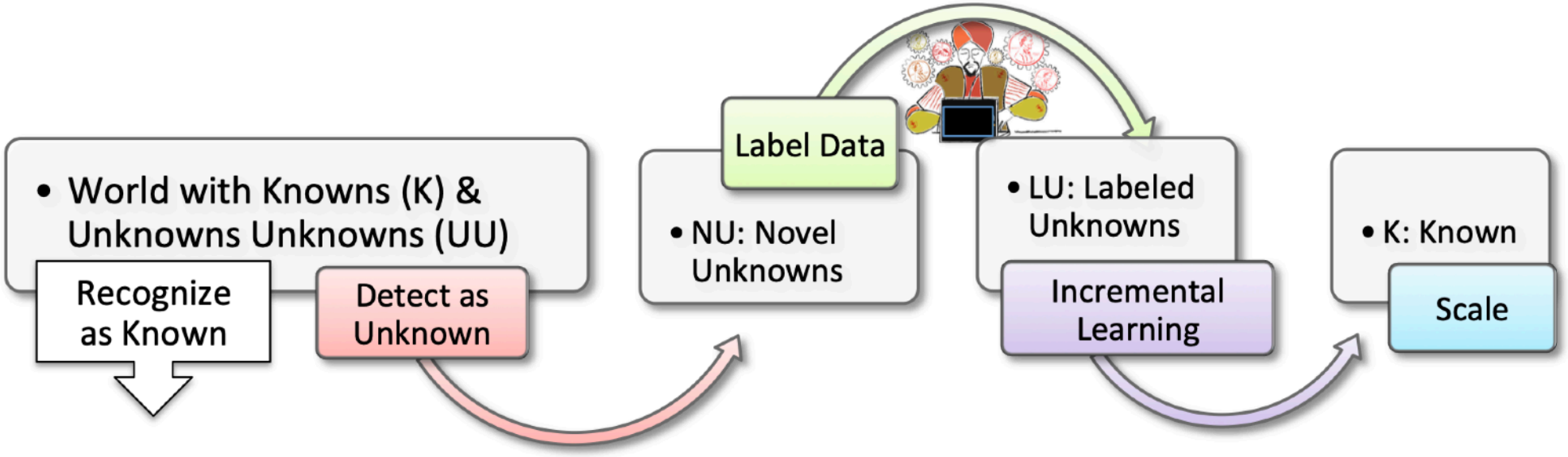


Figure from CVPR16 “Statistical Methods for Open Set Recognition” by Scheirer & Boult, https://www.wjscheirer.com/misc/openset/cvpr2016-open-set-part3.pdf

A (slow) trend towards open world learning?

"An effective open world recognition system must efficiently perform four tasks: detect unknown, choose which points to label for addition to the model, label the points, and update the model"

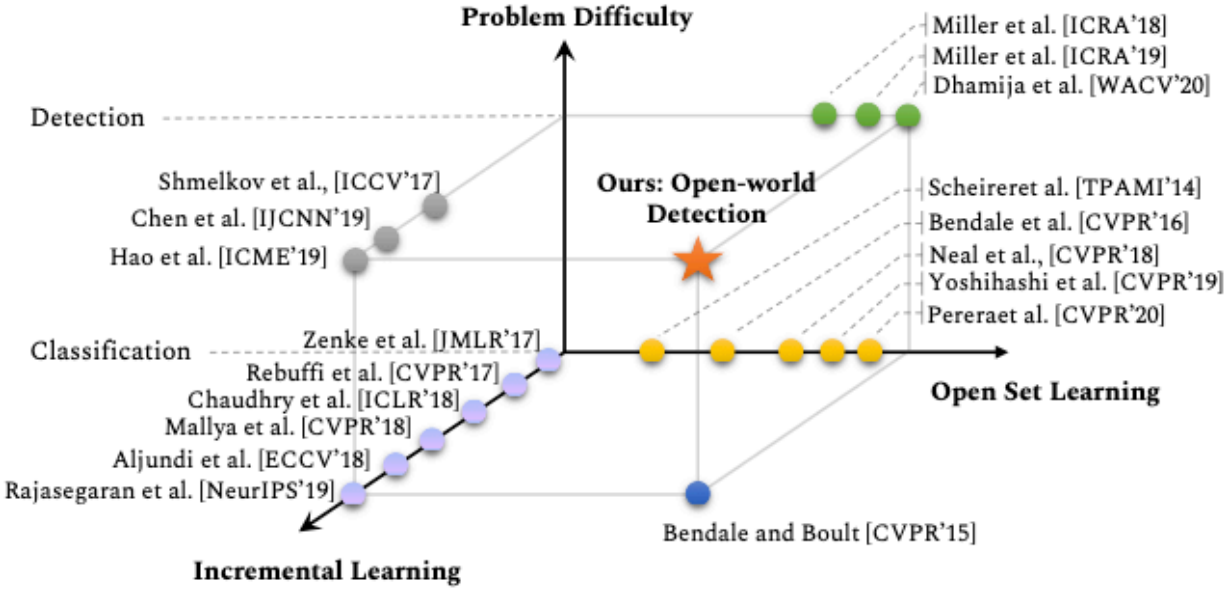


Boult et al, "Learning and the Unknown", AAAI 2019

A (slow) trend towards open world learning?

Stationary + closed world benchmarks have been the traditional focus of ML!

Open world is challenging, but critical to any lifelong learner



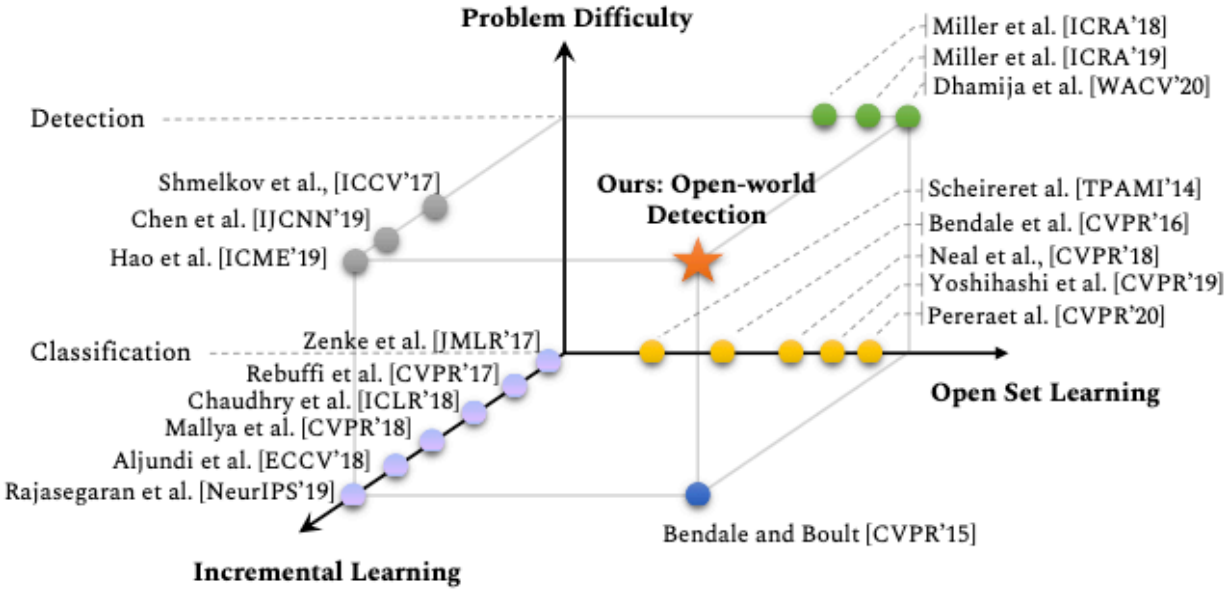
Joseph et al, "Towards Open World Object Detection", CVPR 2021

A (slow) trend towards open world learning?

Stationary + closed world benchmarks have been the traditional focus of ML!

Open world is challenging, but critical to any lifelong learner

Moving beyond closed sets is often ignored in “continual” ML



With modern large models it is receiving increasing attention again, e.g. in “open vocabulary learning”, “hallucinations” etc.

Joseph et al, “Towards Open World Object Detection”, CVPR 2021

Universität Bremen

OWL ML

You've made it through the course! The "only" thing left to do is to summarize & connect the dots!

We have learned about a lot of important concepts!
How do they interact & how can they be combined?

We learned: more to ML than “train-val-test”

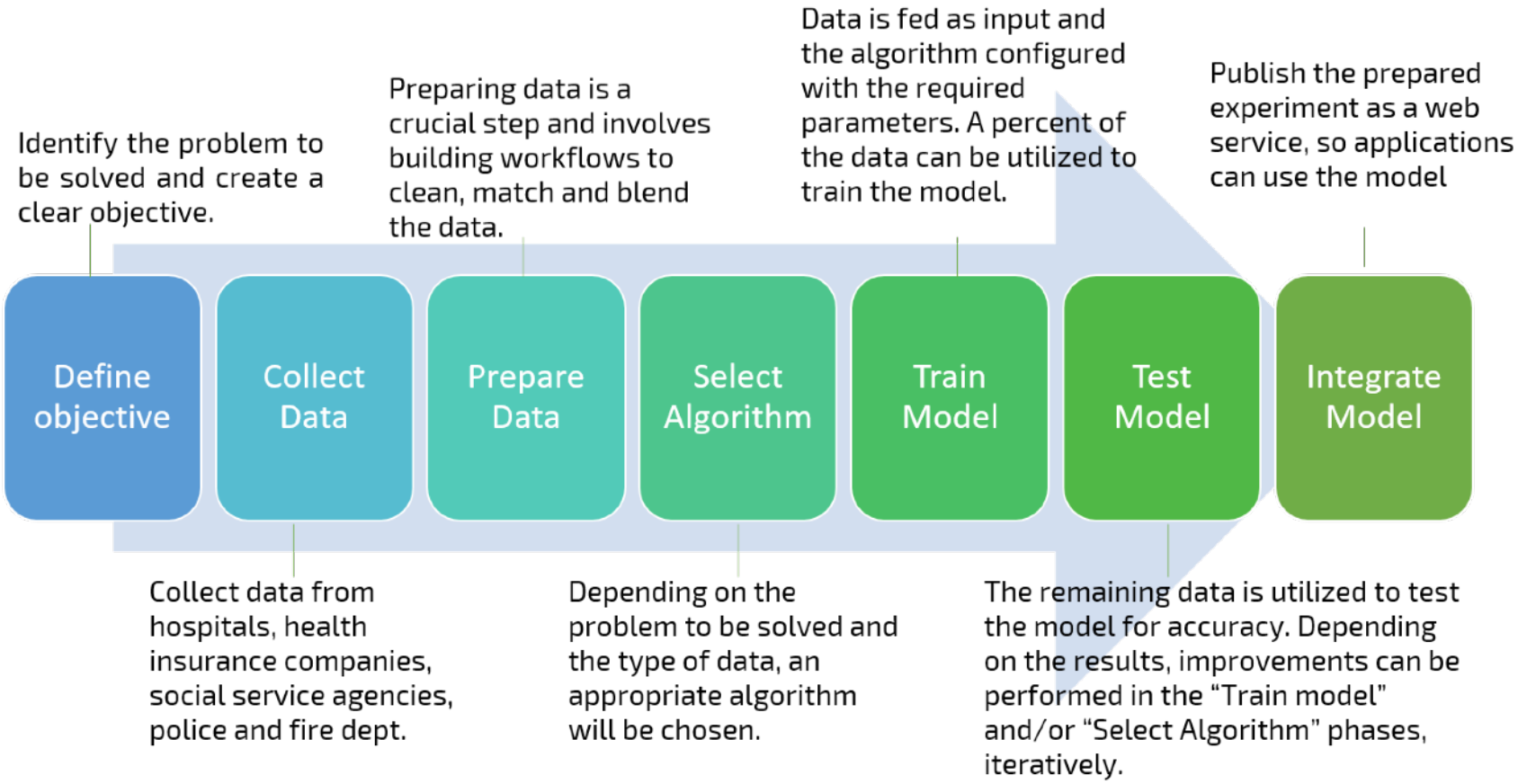
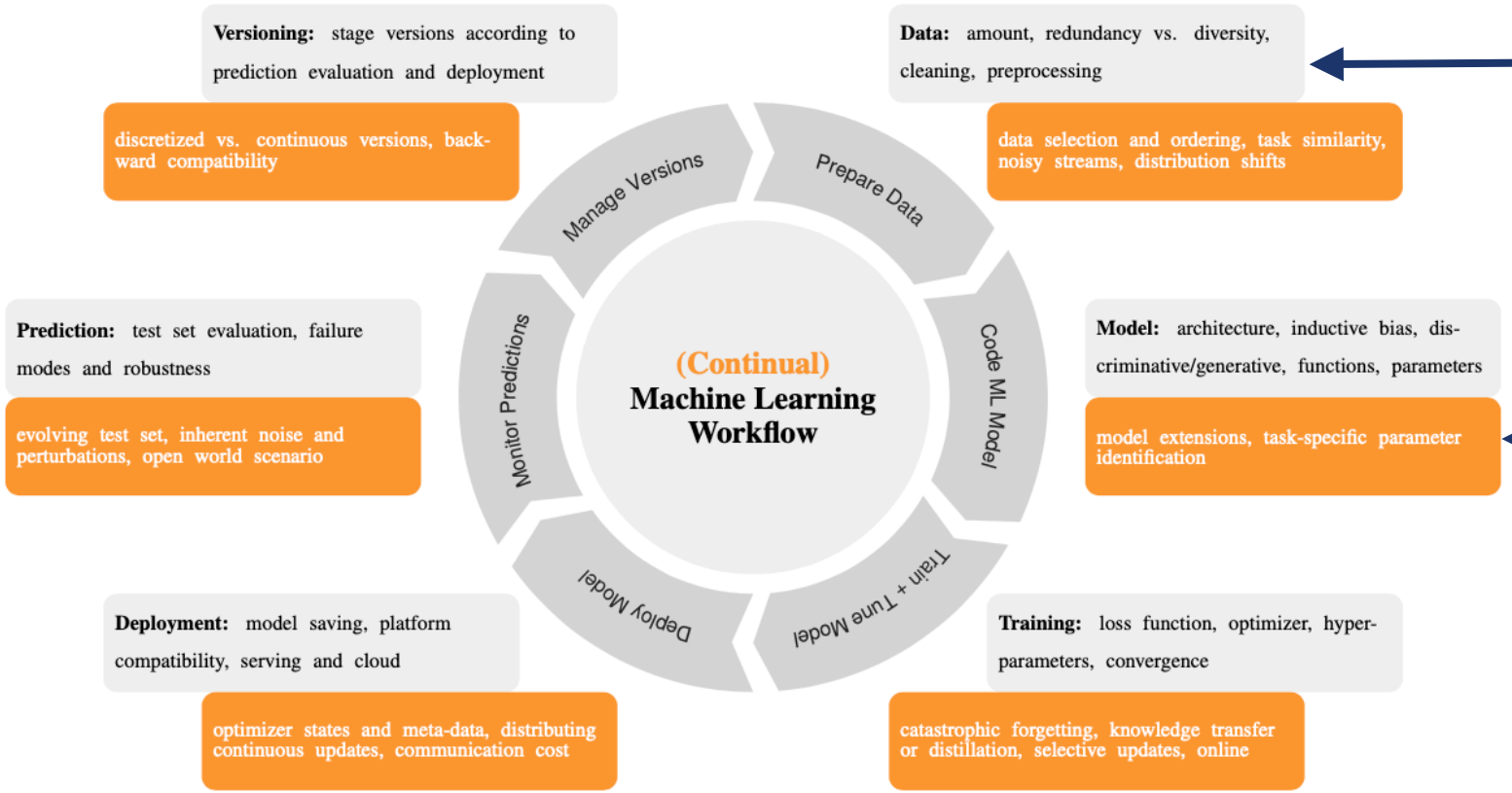


Figure from https://www.congrelate.com/get-workflow-machine-learning-images/

We learned: lifelong ML has many challenges



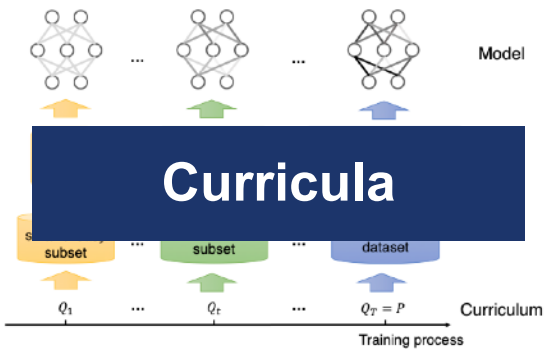
"Stationary" individual questions

"Non-stationary" extra dependencies and synergies

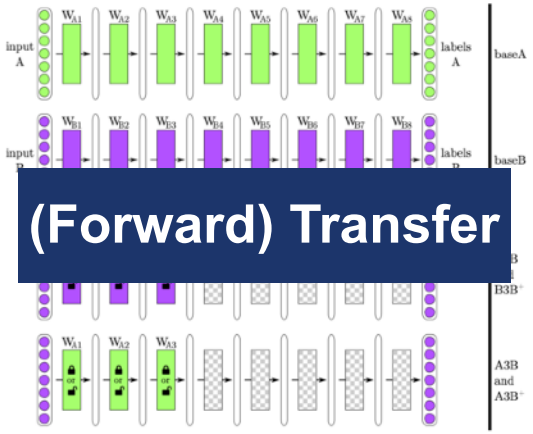
Figure from Mundt et al, "CLEVA-Compass: A Continual Learning Evaluation Assessment Compass to Promote Research Transparency and Comparability", ICLR 2022

Universität Bremen

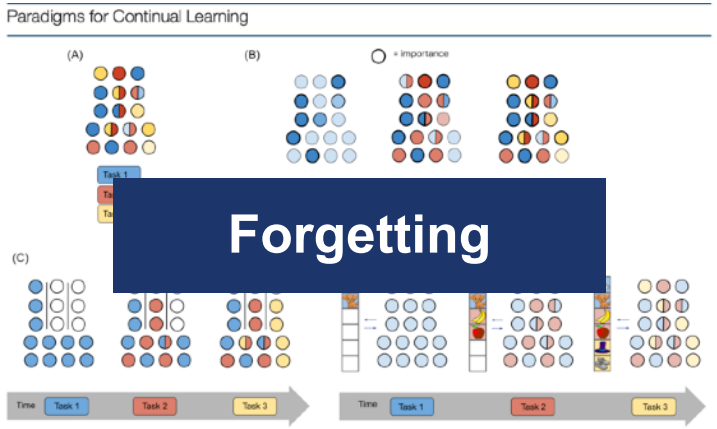
What we have learned in detail



Curricula



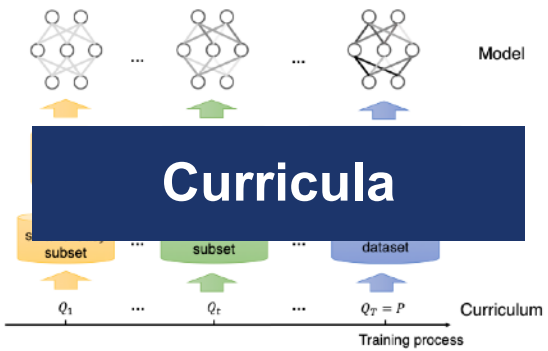
(Forward) Transfer



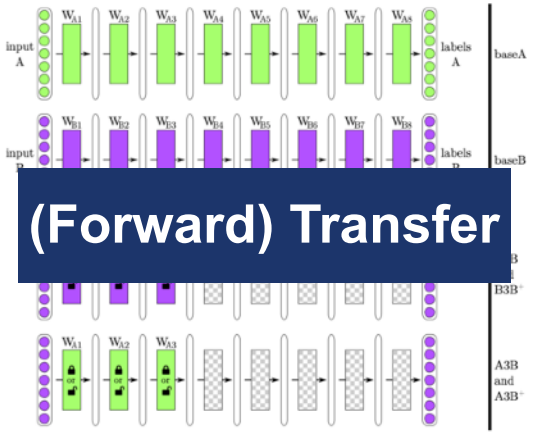
Forgetting

Universität Bremen

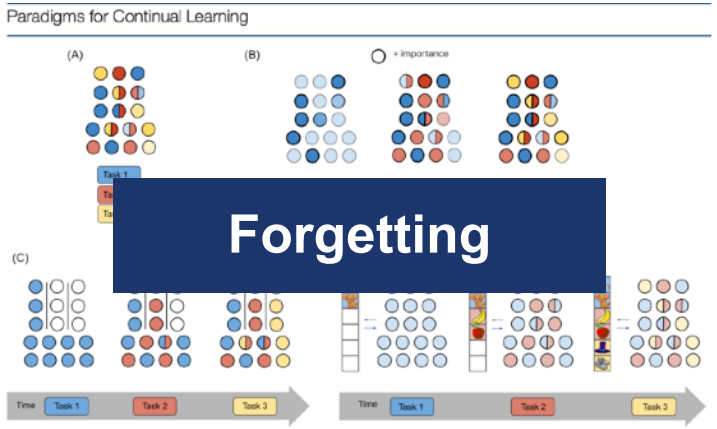
What we have learned in detail



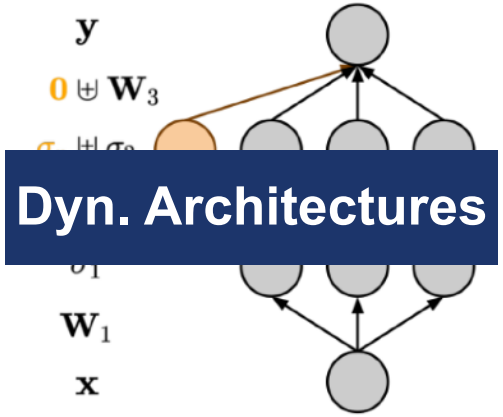
Curricula



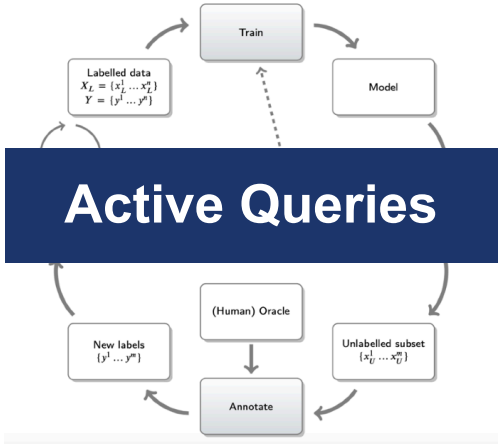
(Forward) Transfer



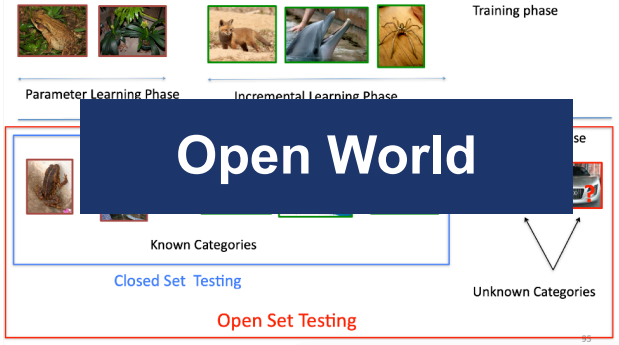
Forgetting



Dyn. Architectures

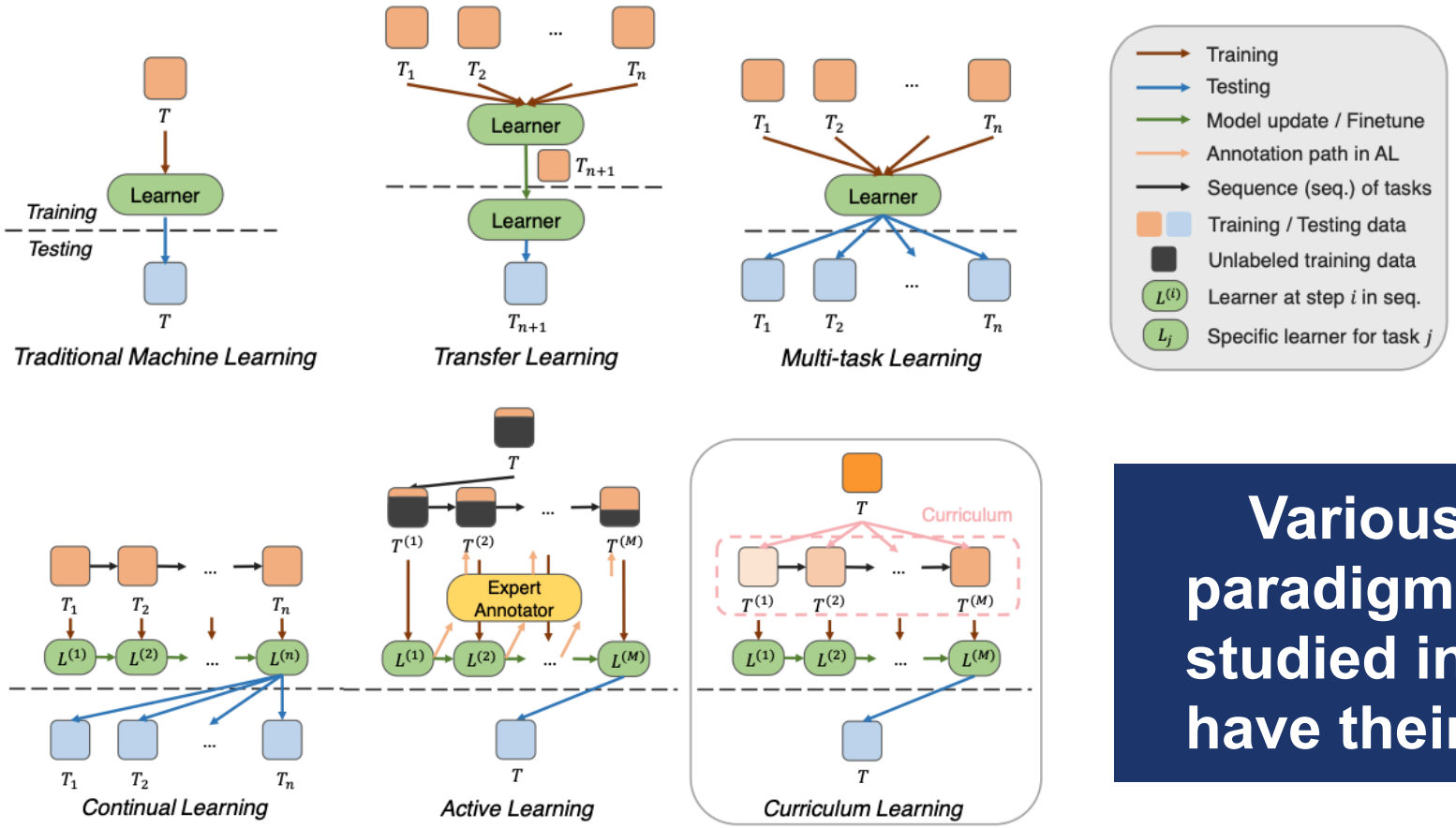


Active Queries



Open World

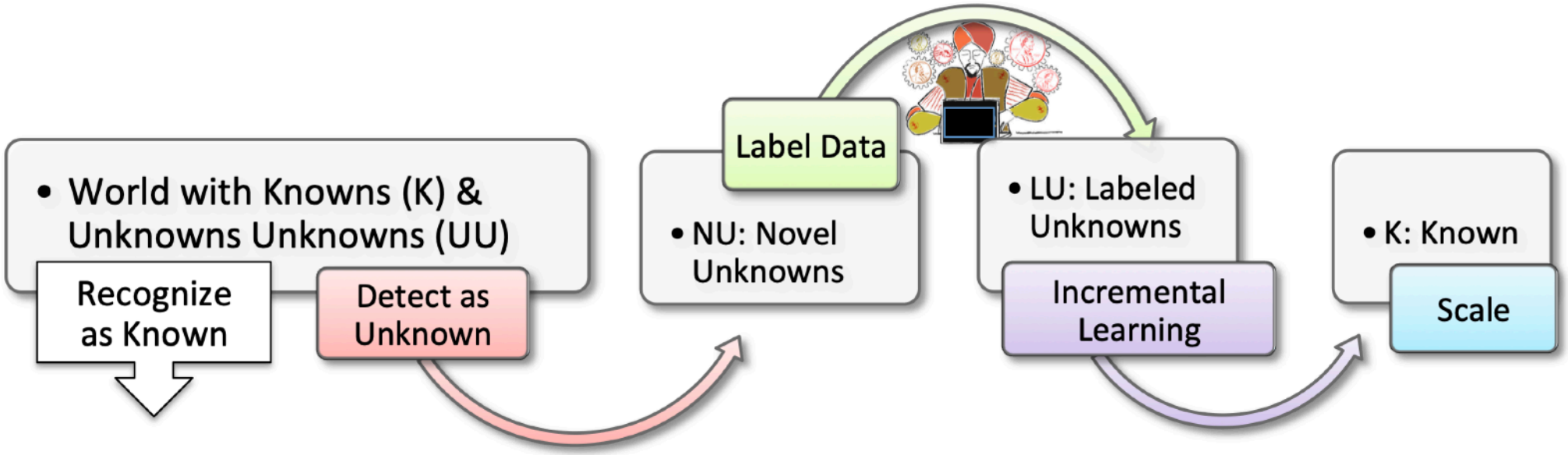
What we have learned in detail



Various "individual" paradigms that are often studied in isolation each have their own frontiers!

Wang et al, "A Survey on Curriculum Learning", TPAMI 2021

What we have learned in detail



But they are still puzzle pieces to a greater whole that we still haven't researched sufficiently yet to have figured out "lifelong ML"

Bendale & Boult ,"Towards Open World Recognition", CVPR 2015

Early definition of lifelong ML

Definition - Lifelong Machine Learning - Thrun 1996:
"The system has performed N tasks. When faced with the (N+1)th task, it uses the knowledge gained from the N tasks to help the (N+1)th task."

We learned this definition is more akin to a (still incomplete) description of transfer learning

"Is Learning The n-th Thing Any Easier Than Learning the First?" (NeurIPS 1996) & "Explanation based Neural Network Learning A Lifelong Learning Approach", Springer US, 1996

A slightly newer definition of lifelong ML

Definition - Lifelong Machine Learning - Chen & Liu 2017:

"Lifelong Machine Learning is a continuous learning process. At any time point, the learner performed a sequence of N learning tasks, $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N$ .These tasks can be of the same type or different types and from the same domain or different domains. When faced with the (N+1)th task $\mathcal{T}_{N+1}$ (which is called the new or current task) with its data D_{N+1}, the learner can leverage past knowledge in the knowledge base (KB) to help learn $\mathcal{T}_{N+1}$. The objective of LML is usually to optimize the performance on the new task $\mathcal{T}_{N+1}$, but it can optimize any task by treating the rest of the tasks as previous tasks. KB maintains the knowledge learned and accumulated from learning the previous task. After the completion of learning $\mathcal{T}_{N+1}$, KB is updated with the knowledge (e.g. intermediate as well as the final results) gained from learning $\mathcal{T}_{N+1}$. The updating can involve inconsistency checking, reasoning, and meta-mining of additional higher-level knowledge."

"Lifelong Machine Learning", Chen & Liu, Morgan Claypool, 2017

A slightly newer definition of lifelong ML

Definition - Lifelong Machine Learning - Chen & Liu 2017:
"Lifelong Machine Learning is a continuous learning process. At any time point, the learner performed a sequence of N learning tasks $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N$. These tasks can

May contain some parts we haven't discussed in the course: reasoning, meta-mining of higher-level knowledge …

Does not explicitly contain many things we have learned about: active data queries, difficulty/curricula, dynamic model architectures, open worlds, memory/compute constraints …

knowledge (e.g. intermediate as well as the final results) gained from learning $\mathcal{T}_{N+1}$. The updating can involve inconsistency checking, reasoning, and meta-mining of additional higher-level knowledge."

"Lifelong Machine Learning", Chen & Liu, Morgan Claypool, 2017

Universität Bremen

A slightly newer definition of lifelong ML

Definition 4. Continual Machine Learning - this work: The learner performs a sequence of N continual learning tasks, $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N$, that are distinct from each other in terms of shifts in the underlying data distribution. The latter can imply a change in objective, transitions between different domains or inclusion of new modalities. At any point in time, the learner must be able to robustly identify unseen unknown data instances. Depending on what is permissible in application contexts, the learner can either reject such instances in a non-controllable data stream or set them aside for later learning. In the latter scenario, the learner should be able to rank order unknowns according to similarity with existing tasks, in order to actively build a meaningful learning curriculum itself. If the system is desired to be supervised, a human in the loop may group and label the set of identified unseen unknowns to explicitly guide future learning. When faced with a selected $(N{+}1)$th task $\mathcal{T}_{N+1}$ (which is called the new or current task) with its data $\mathcal{D}_{N+1}$, the learner should leverage its dictionary of representations to accelerate learning of $\mathcal{T}_{N+1}$ (forward transfer), extend the dictionary with unique representations obtained from the new task's data (this can be completely new types of dictionary elements), while simultaneously maintaining and improving the existing representational dictionary with respect to former tasks (backward transfer).

"A Wholistic View of Continual Learning with Deep Neural Networks", Mundt et al, Neural Networks 160, 2023

Universität Bremen

A slightly newer definition of lifelong ML

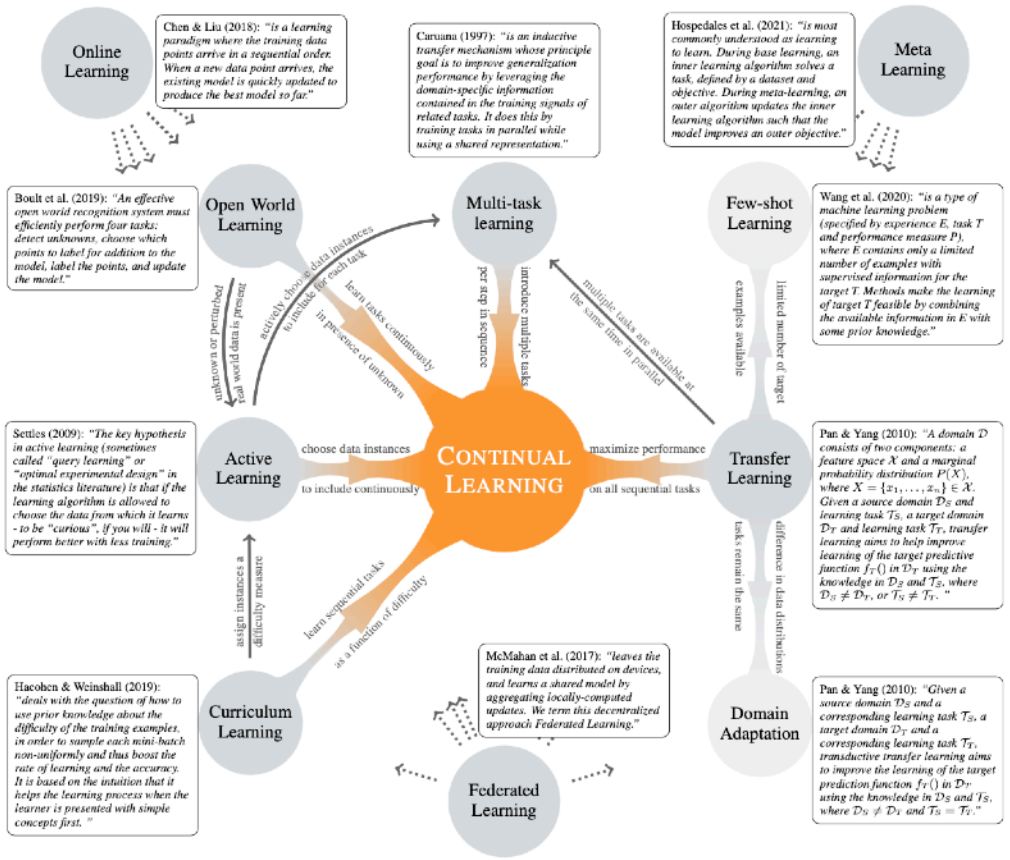
Definition 4. Continual Machine Learning - this work: The learner performs a sequence of N continual learning tasks, $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N$, that are distinct from each other in terms of shifts in the underlying data distribution. The latter can imply a change in objective, transitions between different domains or inclusion of new modalities. At any point in time, the learner must be able to robustly identify unseen unknown data instances. Depending on what is permissible in application contexts, the learner can either reject such instances in a non-controllable data stream or set them aside for later learning. In the latter scenario, the learner should be able to rank order unknowns according to similarity with existing tasks, in order to actively build a meaningful learning curriculum itself. If the system is desired to be supervised, a human in the loop may group and label the set of identified unseen unknowns to explicitly guide future learning. When faced with a selected $(N{+}1)$th task $\mathcal{T}_{N+1}$ (which is called the new or current task) with its data $\mathcal{D}_{N+1}$, the learner should leverage its dictionary of representations to accelerate learning of $\mathcal{T}_{N+1}$ (forward transfer), extend the dictionary with unique representations obtained from the new task's data (this can be completely new types of dictionary elements), while simultaneously maintaining and improving the existing representational dictionary with respect to former tasks (backward transfer).

Now contains curricula, active, open world… omits other parts

Our "definitions" are getting more and more extensive

"A Wholistic View of Continual Learning with Deep Neural Networks", Mundt et al, Neural Networks 160, 2023

A slightly newer definition of lifelong ML



Now contains curricula, active, open world… omits other parts

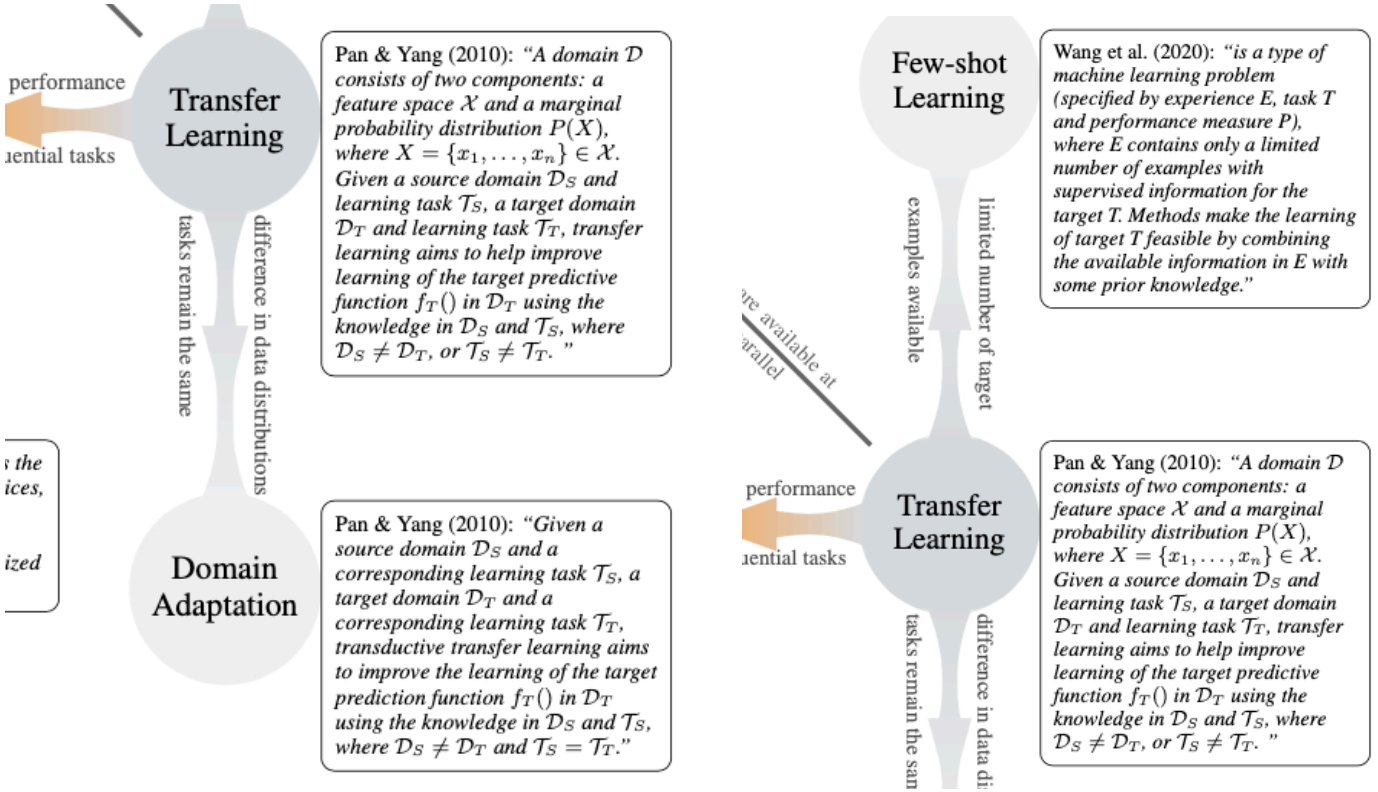
Our "definitions" are getting more and more extensive

The paradigm differences can be nuances, how we evaluate…

But assumptions can be valid in context & express preferences!

Figure from Mundt et al, "CLEVA-Compass: A Continual Learning Evaluation Assessment Compass to Promote Research Transparency and Comparability", ICLR 2022

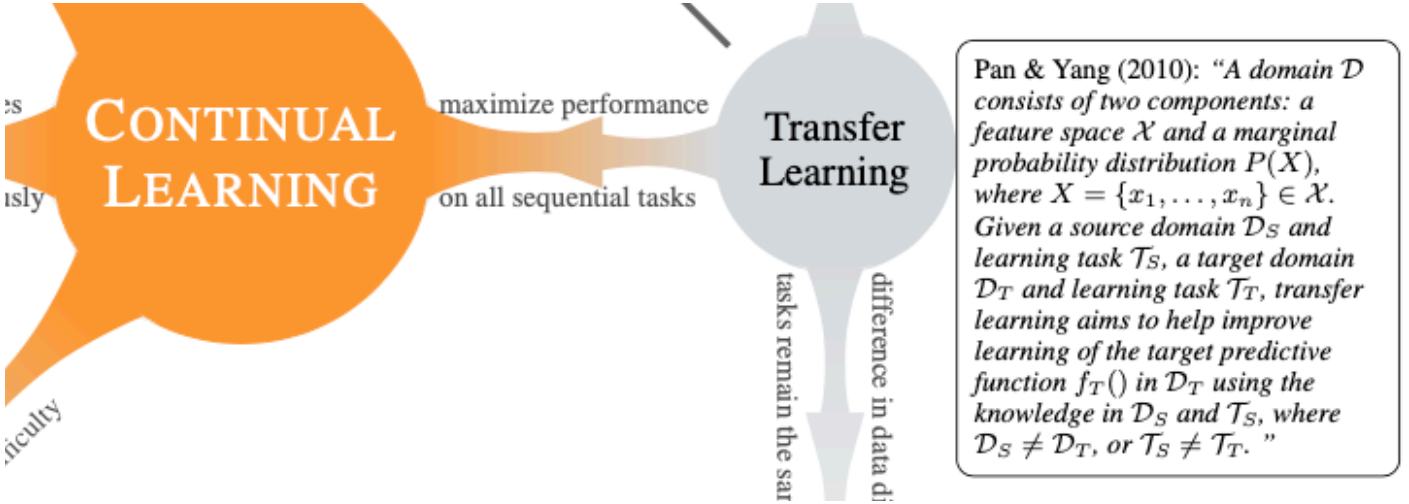
A slightly newer definition of lifelong ML



Transductive, inductive & few shot learning are all variants of (forward) transfer oriented learning

Figure from Mundt et al, "CLEVA-Compass: A Continual Learning Evaluation Assessment Compass to Promote Research Transparency and Comparability", ICLR 2022

A slightly newer definition of lifelong ML

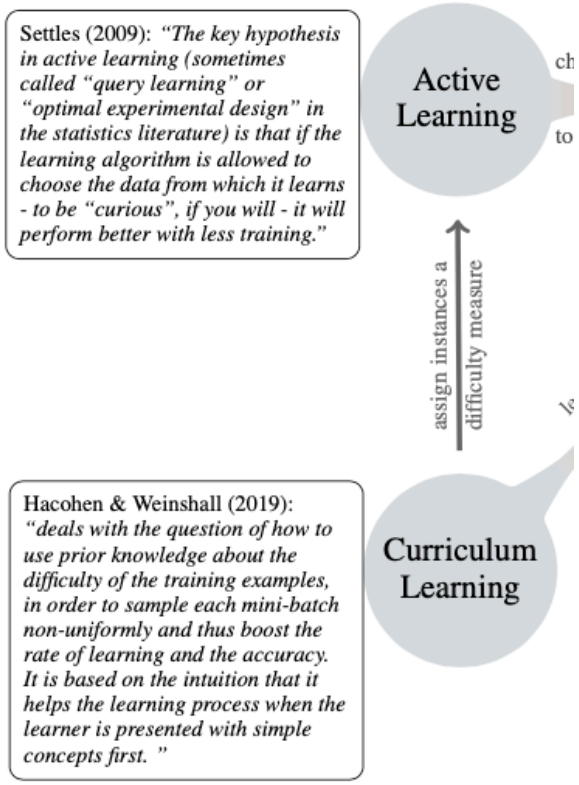


Pan & Yang (2010): *"A domain $\mathcal{D}$ consists of two components: a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$, where $X = \{x_1, \ldots, x_n\} \in \mathcal{X}$. Given a source domain $\mathcal{D}_S$ and learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and learning task $\mathcal{T}_T$, transfer learning aims to help improve learning of the target predictive function $f_T()$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$."*

Continual learning (in a traditional definition) is similar to transfer with more than 2 tasks & avoiding forgetting

Figure from Mundt et al, "CLEVA-Compass: A Continual Learning Evaluation Assessment Compass to Promote Research Transparency and Comparability", ICLR 2022

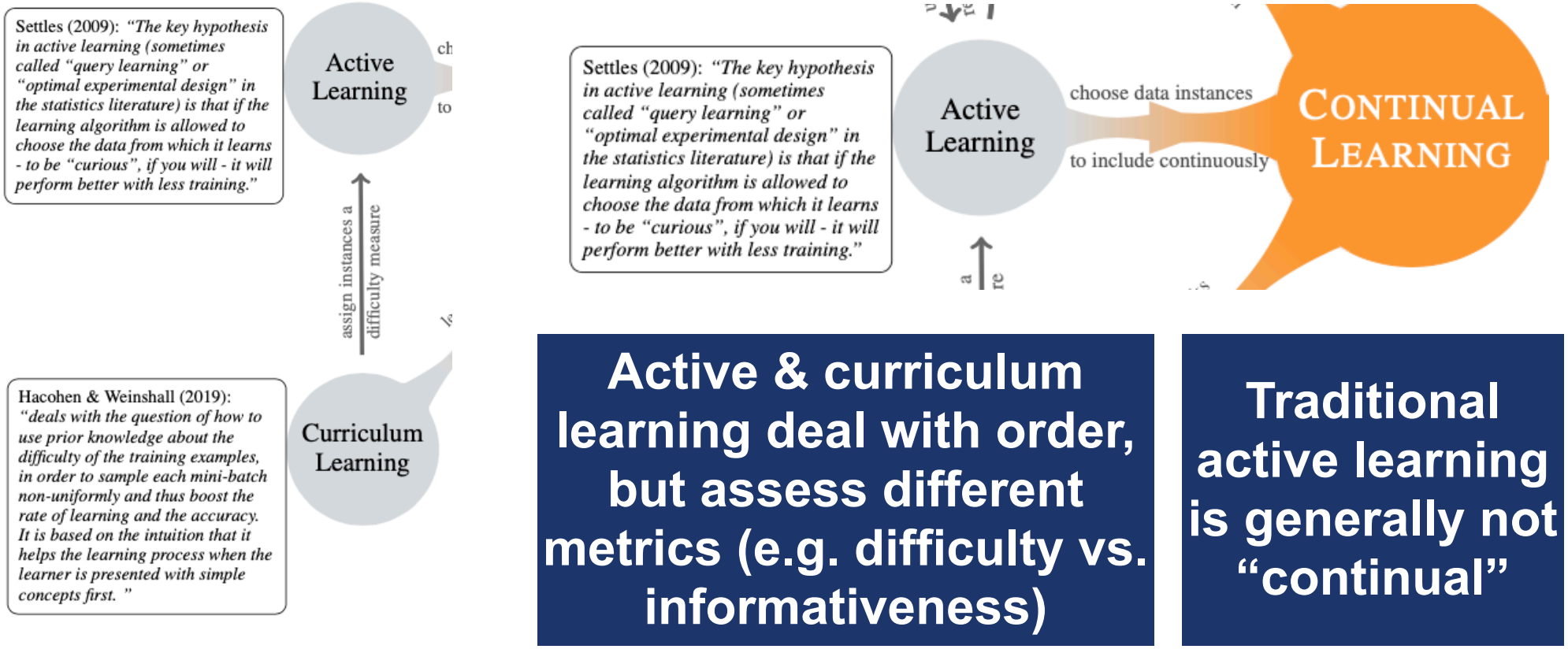
A slightly newer definition of lifelong ML



Active & curriculum learning deal with order, but assess different metrics (e.g. difficulty vs. informativeness)

Figure from Mundt et al, "CLEVA-Compass: A Continual Learning Evaluation Assessment Compass to Promote Research Transparency and Comparability", ICLR 2022

A slightly newer definition of lifelong ML

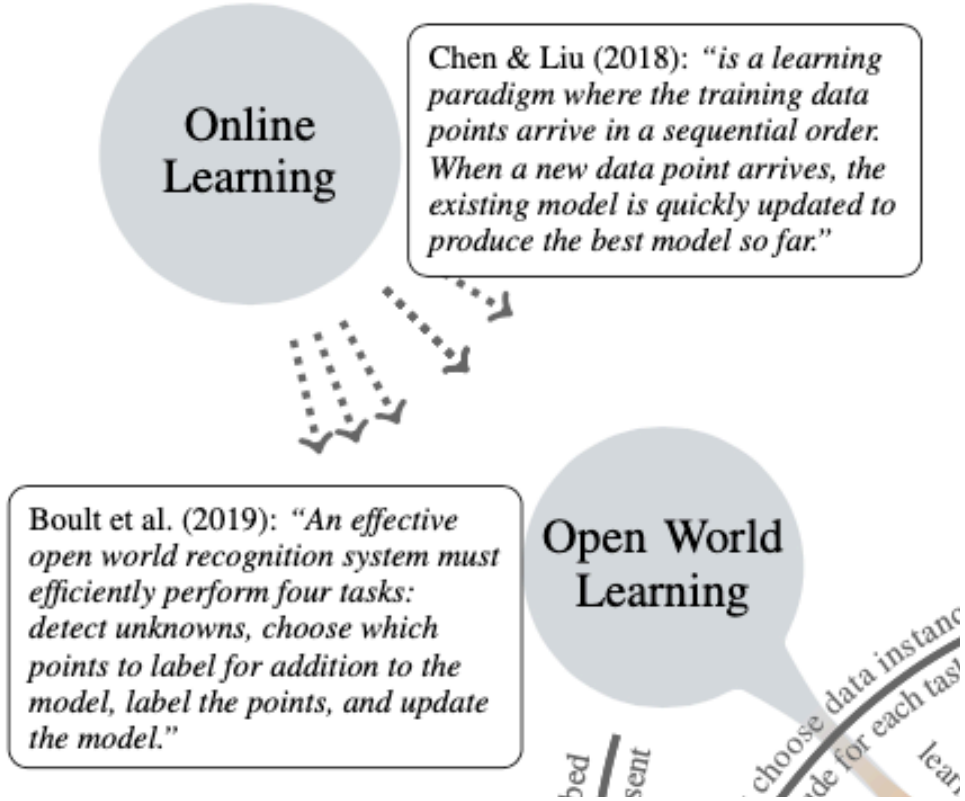


Active & curriculum learning deal with order, but assess different metrics (e.g. difficulty vs. informativeness)

Traditional active learning is generally not "continual"

Figure from Mundt et al, "CLEVA-Compass: A Continual Learning Evaluation Assessment Compass to Promote Research Transparency and Comparability", ICLR 2022

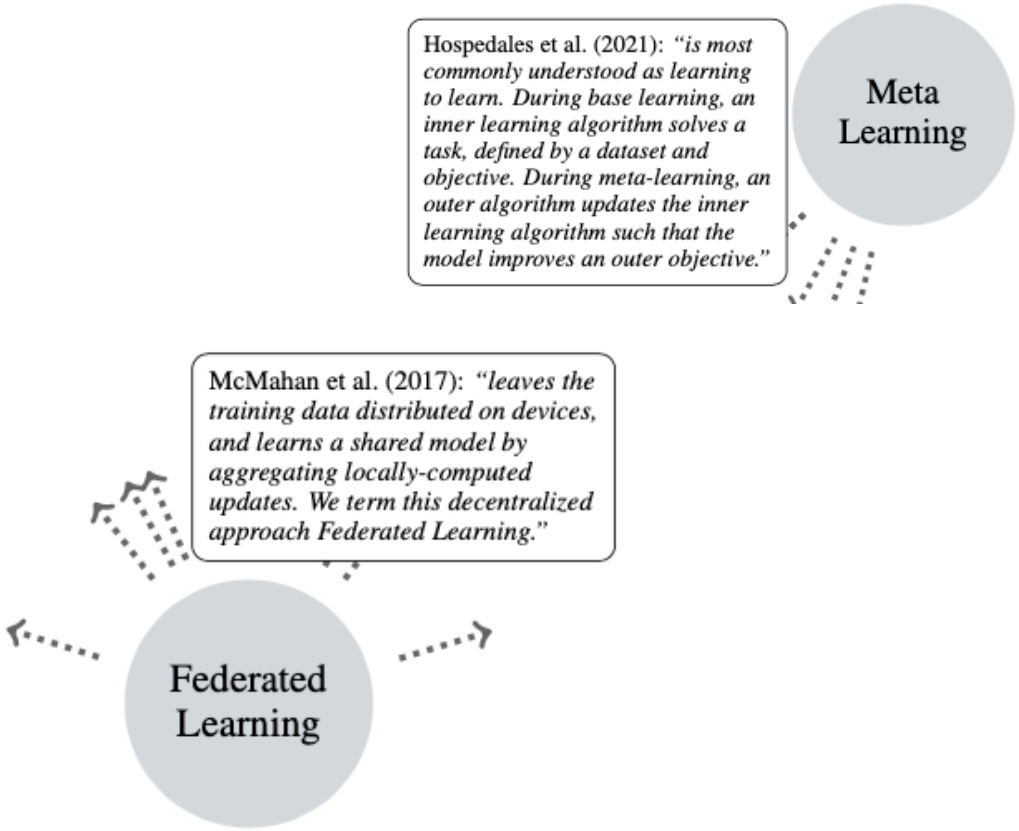
A slightly newer definition of lifelong ML



Open world & online learning break common (deep) ML assumptions, that are often even upheld in ML paradigms focused on non-stationary settings

Figure from Mundt et al, "CLEVA-Compass: A Continual Learning Evaluation Assessment Compass to Promote Research Transparency and Comparability", ICLR 2022

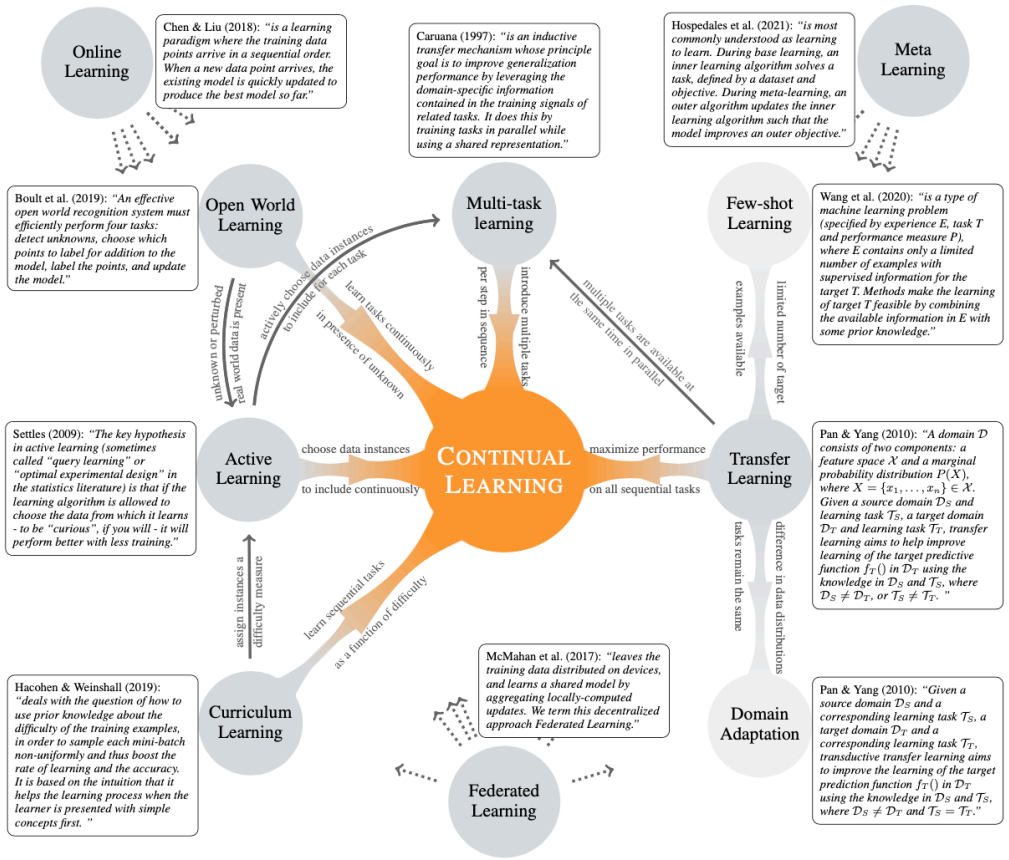
A slightly newer definition of lifelong ML



Federated/distributed learning & meta-learning aren't yet fully treated within a "lifelong learning" scope, but share some critically similar technical challenges

Figure from Mundt et al, "CLEVA-Compass: A Continual Learning Evaluation Assessment Compass to Promote Research Transparency and Comparability", ICLR 2022

A slightly newer definition of lifelong ML

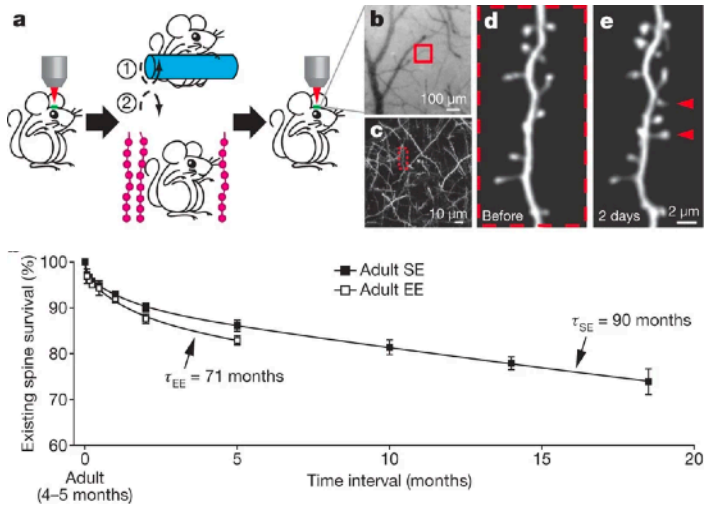


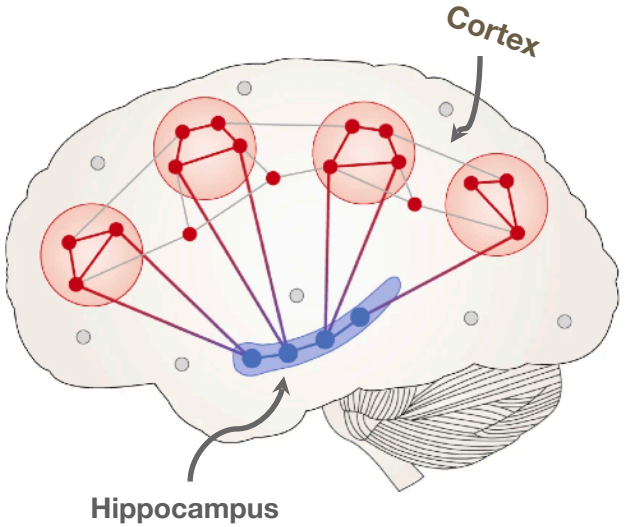
My subjective opinion: overall it may not be best goal to "define" a single continual/ lifelong learning ML paradigm

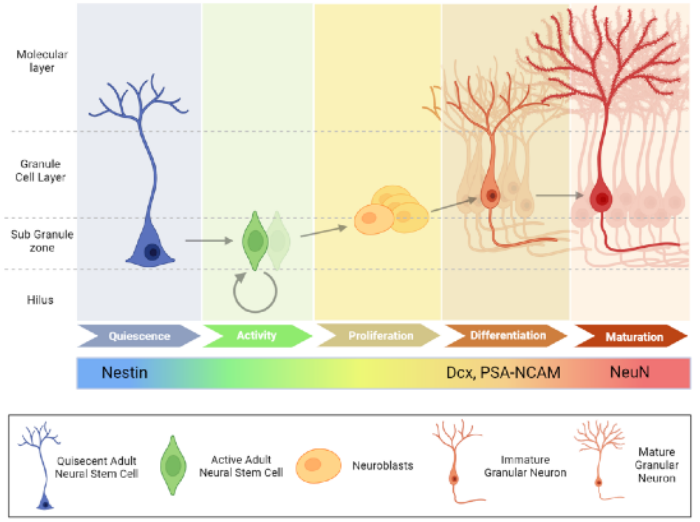
But there are key capabilities that are necessary in many realistic set-ups & lots of evidence that many mechanisms contribute!

Figure from Mundt et al, "CLEVA-Compass: A Continual Learning Evaluation Assessment Compass to Promote Research Transparency and Comparability", ICLR 2022

Biological evidence points to many sources







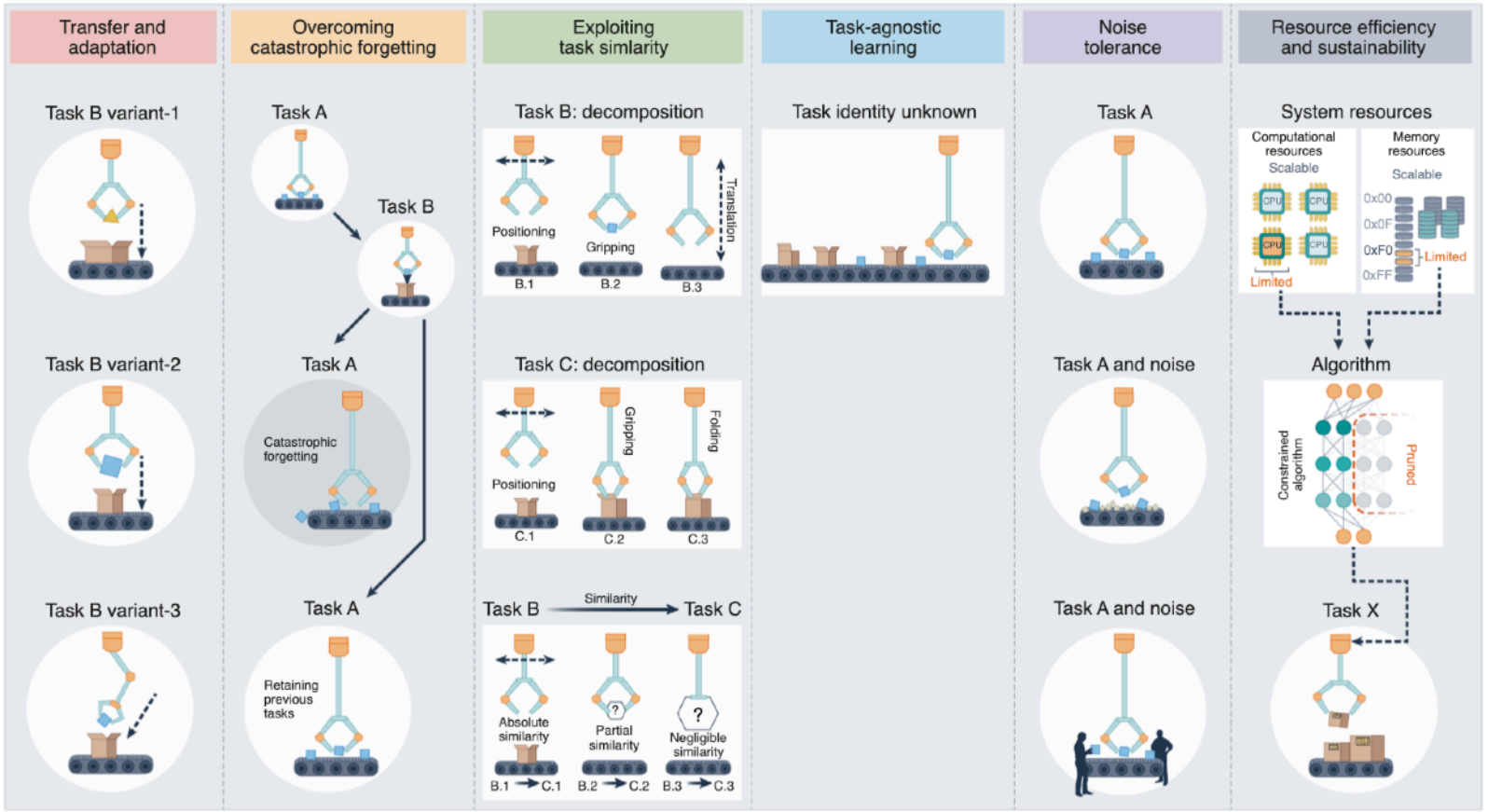
Regularization of dendritic spines

Complementary learning systems theory

(Adult) neurogenesis

Left figure adapted from Cichon & Gan 2015, Center figure adapted from https://www.biorender.com/template/adult-neurogenesis, Right figure adapted from Klinzing et al., 2019

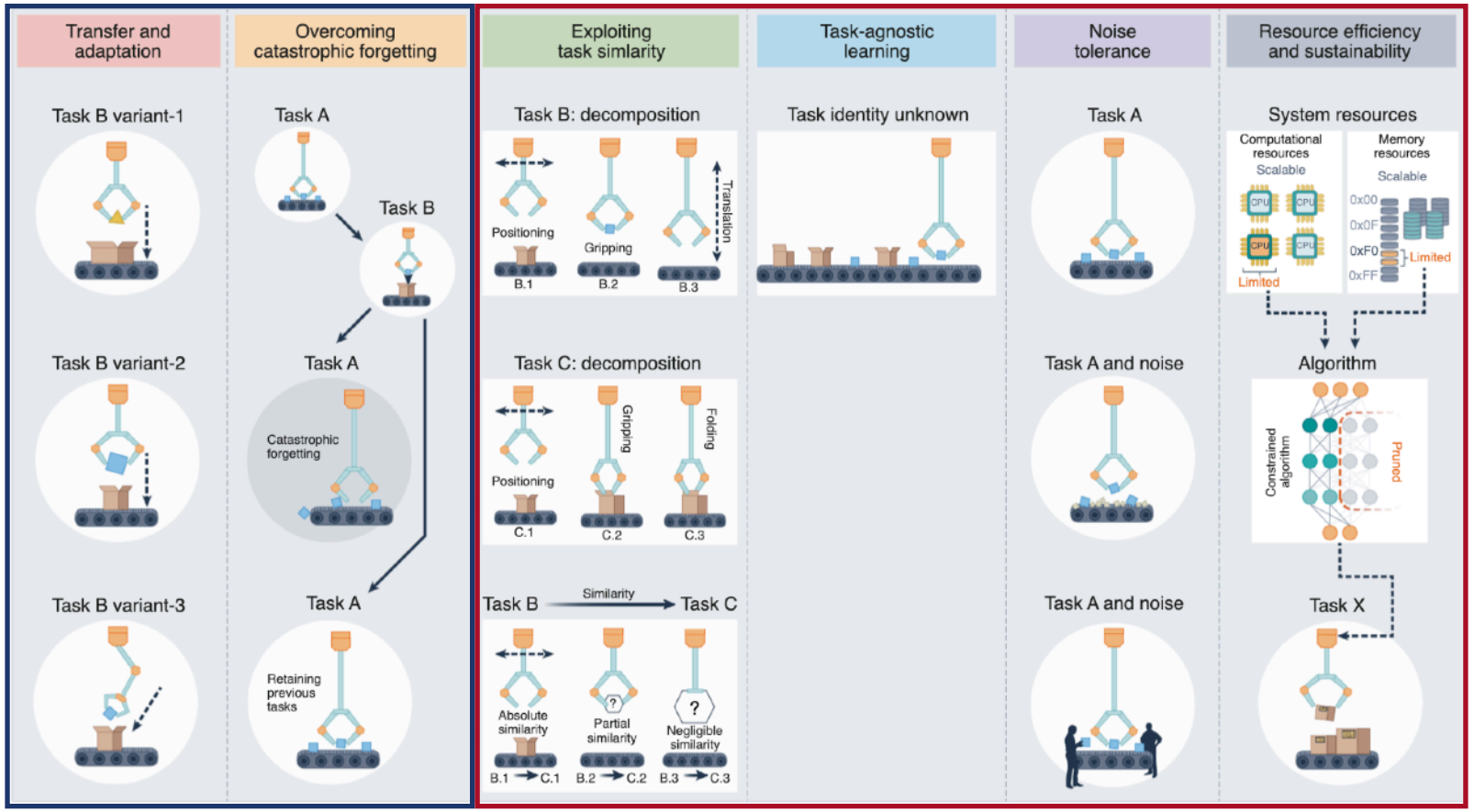
And not just in order to avoid forgetting!



Kudithipudi et al, “Biological underpinnings for lifelong learning machines”, Nature Machine Intelligence (4), 2022

And not just in order to avoid forgetting!

The typical current focus



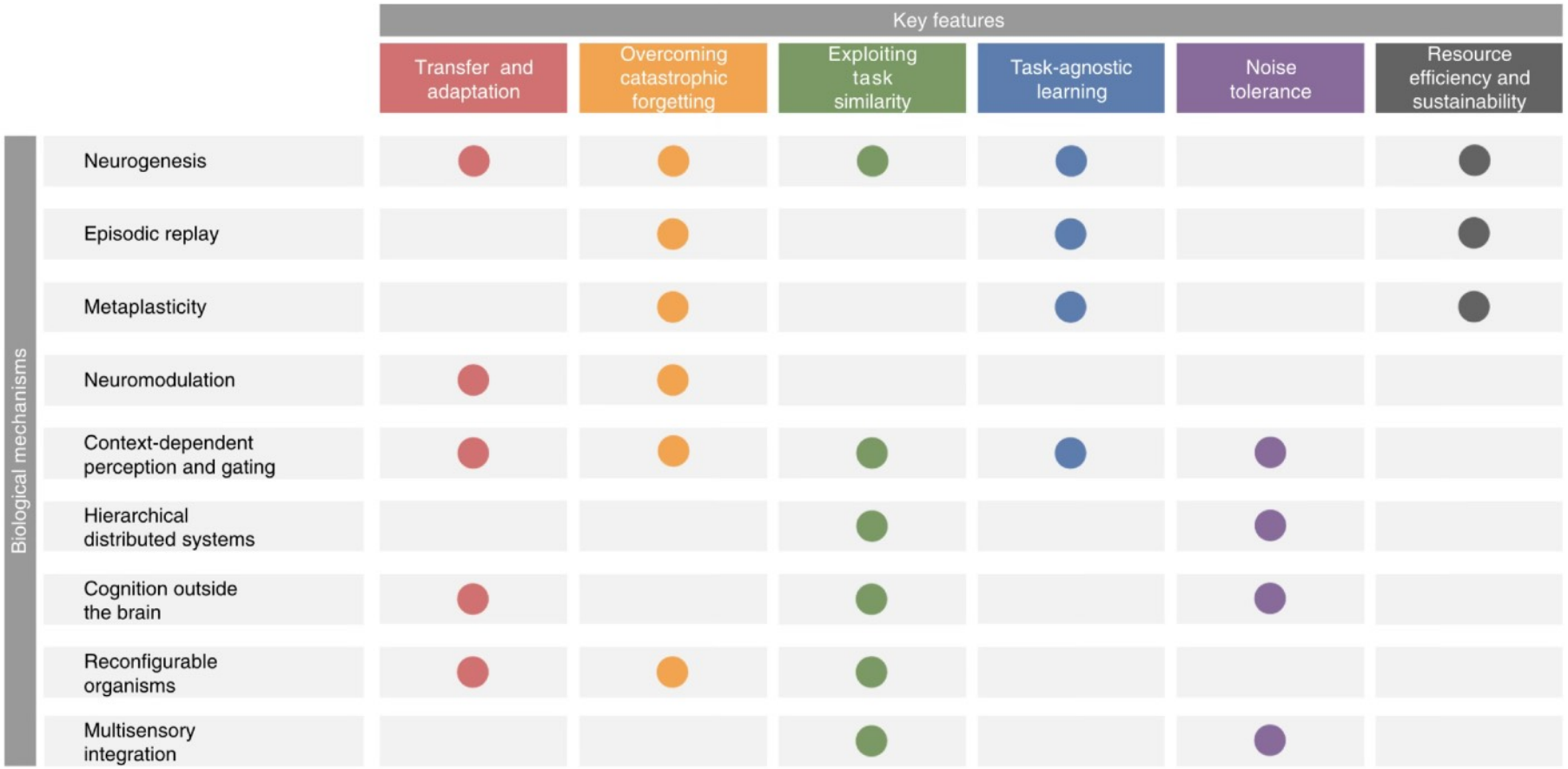
The less researched (so far)

Your chance to shine & define the future of ML!

Kudithipudi et al, “Biological underpinnings for lifelong learning machines”, Nature Machine Intelligence (4), 2022

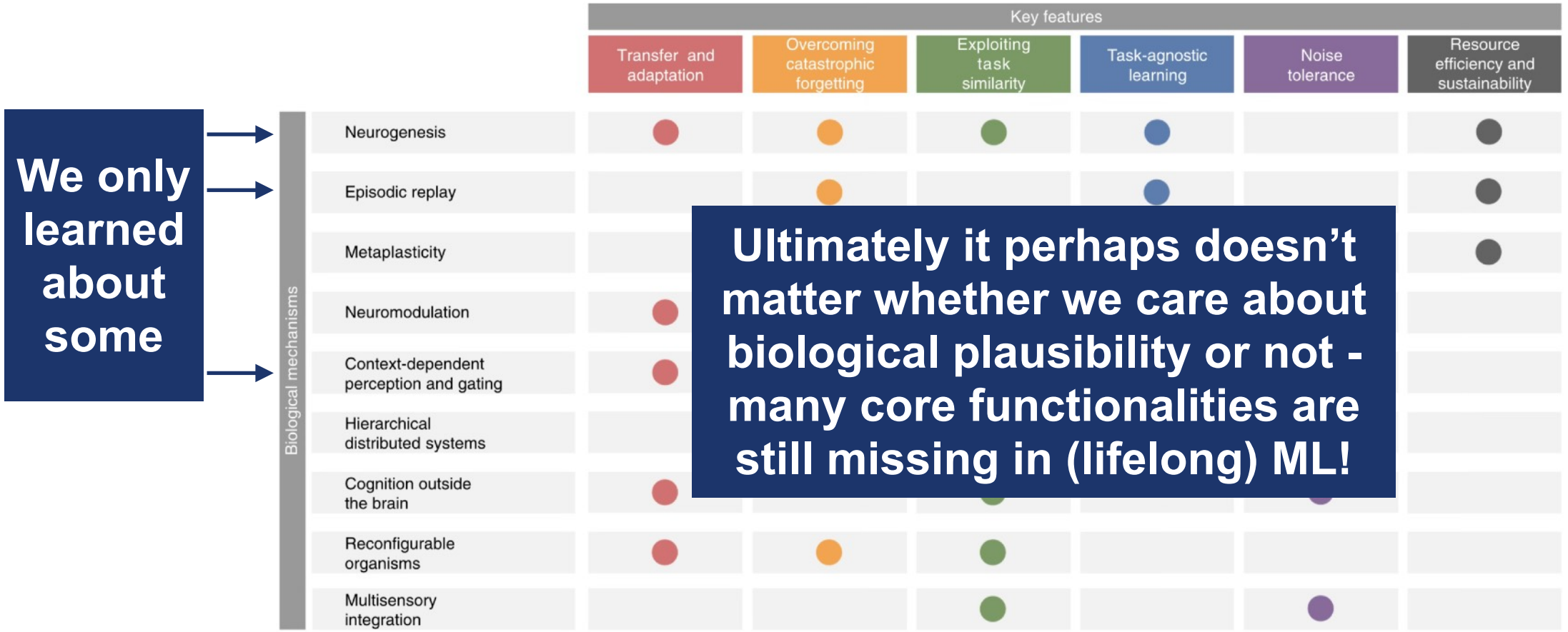
With many complementary mechanisms

We only learned about some



Kudithipudi et al, “Biological underpinnings for lifelong learning machines”, Nature Machine Intelligence (4), 2022

With many complementary mechanisms



Kudithipudi et al, "Biological underpinnings for lifelong learning machines", Nature Machine Intelligence (4), 2022

Universität Bremen

Thank you for attending the course!

If you are interested in a thesis/guided research, come work with us to advance some of the learned concepts:

- *Probabilistic & generative ML*
- *Mitigating (catastrophic) forgetting in ML*
- *Knowledge in deep networks & its distillation*
- *Dynamically composable neural architectures*
- *Overconfidence, learning shortcuts, open worlds*

OWL-ML: Open World Lifelong Machine Learning
Cartesium: http://owl-ml.uni-bremen.de